A MICROCOMPUTER PROGRAM FOR INVENTORY CONTROL AND BILLING FOR A UNIVERSITY FEED MANUFACTURING FACILITY

by

Coree Lynn Smith

B. S., Kansas State University, 1981

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Animal Science and Industry

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1984

Approved by:

B. E. Brent

Major Professor

LD
2668
.R4
1984
.S64
c. 2

A11202 666633

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation and utmost gratitude to all those who provided their personal assistance for the completion of this study.

Grateful appreciation is expressed to Dr. B. E. Brent, my major professor, for his valuable guidance and help during this study. Special appreciation is expressed to the members of the advisory committee, Dr. Roger Terry, and Dr. Steve Pollman, for their assistance and guidance.

An appreciation is expressed to my family who gave their encouragment, love, and inspiration during the past six years of study.

Last, but not least a special thank you is given to Daniel Oakleaf my fiance' and future husband, for his love, encouragement, help, and patience during the last year of my academic career.

THIS BOOK CONTAINS NUMEROUS PAGES WITH THE ORIGINAL PRINTING BEING SKEWED DIFFERENTLY FROM THE TOP OF THE PAGE TO THE BOTTOM.

THIS IS AS RECEIVED FROM THE CUSTOMER.

TABLE OF CONTENTS

Chapter		Page
I.	INTRODUCTION	
	A. Background	1
	B. Overview	2
II.	USER GUIDE	
	A. Getting started	3
	B. Diskettes and disk drives	3
	C. Purpose	3
	D. System capacitities	4
	E. General operating information	4
	F. Problem solving procedure	6
	G. The menu choices in more detail	8
III.	TECHNICAL GUIDE	
	A. Introduction to Technical Guide	27
	B. File type definitions	27
	C. Special features	27
	MAIN PROGRAM	
	D. File information	28
	E. Major components	29
	F. Variables and arrays	31
	G. Description of individual program components	37
	REPORTS PROGRAM	
	H. Special features	51
	I. File information	51
	J. Major routines	52
	K. Variable and arrays	52
	L. Decription of individual program routines	56
	M. Solution procedure	57
IV.	FUTURE UPDATES	60
V.	LITERATURE CITED	61
APPENDIX		
	Appendix A.1 - Inventory list report	63
	Appendix A.2 - Complete ration listing by unit	64
	Appendix A.3 - Month-to-date purchases	65
	Appendix A.4 - Month-to-date sales	66
	Appendix A.5 - Ration Id and number by unit	67
	Appendix A.6 - Month-to-date and Month-end closeouts ingredient use report	68
	Appendix A.7 - Month-to-date and Month-end closeouts project number use report	69
	Appendix A.8 - Month-to-date and Month-end closeouts ration use report	70
	Appendix A.9 - Alpabetized inventory report	71
	Appendix B.1 - Main program listing	72
	Appendix B.2 - Reports program listing	91
	Appendix C.1 - Main program pseudo-code listing	98
	Appendix C.2 - Reports program pseudo-code listing	102
	Appendix D - Acessing Profile III+ through BASIC	103

I. INTRODUCTION

A. Background

Currently the Kansas State University Animal Sciences and Industry Department's feedmill employs a manual system for processing sales, purchases and inventory. The process consists of hand-calculating the value of each ration, the prices and total for each sale and ingredient purchase, and adjusting inventories. Thus, charging the rations to each research project and/or research unit is slow and labor-intensive.

The program is written in BASIC, using the NEWDOS/80.2 disk operating system (Apparat, 1981); a decision based on software, languages and hardware available in the Department when the program was designed in 1982. Special considerations of the feedmill operation impose several constraints upon the characteristics of the language and operating system that could be used for this job.

The mill prepares well over 100 individual rations. These rations contain from three up to 20 individual ingredients. Thus, in the ration file, if a record is sized for the maximum 20 ingredients, a tremendous amount of space is wasted on the disk when three ingredient rations are used. NEWDOS/80.2 is unique in its ability to support records of variable length within a given file.

A further reason for writing the program in BASIC is that several individuals in the Department are fluent in BASIC, so they can participate in future modifications, updates and other program maintenance.

A relational database, such as dBASE II, along with its accessory language (command files) would have been appropriate for the job, except for two reasons: record lengths within a file are fixed, and it requires the CP/M operating system (Digital Research, 1982). The latter problem was solved with the introduction by Radio Shack of the Model 4 microcomputer in 1983; the either Department has

eight. The Radio Shack Model 4 has the advantage of working with either Radio Shack Model III or CP/M software.

Some characteristics of the program are patterned after a feedlot management program in use at Smith Cattle, Inc., of Tribune, Ks. That program was written by High Plains Computing, of Amarillo, Tx., and is completely menu driven. However, since only five comparatively simple rations are used in the feedyard, that program's ration file is relatively small. However, the High Plains computing program does maintain an inventory list, and what ingredients are available for use.

This Master's report presents a system that is manageable, efficient, provides easy access and updating and stores all relevant data for future use.

B. Overview

Described here is the design and use of a BASIC language microcomputer program for inventory control and billing for a university feed manufacturing facility. It runs on a Radio Shack model III or 4 (in model III mode) with two disk drives and at least 48K of user memory and the NEWDOS/80.2 operating system.

The impetus for the project was the department's need for an efficient, user friendly, responsive and manageable feedmill accounting system that would facilitate distributing feed costs to research projects and animal units.

This report is divided into a users guide and a technical guide. The users guide provides step by step directions for program operation. The technical guide is designed from a programmer's veiwpoint. It includes source code listing, with variables and explanations, and provides information necessary for program maintenance, update, and additions.

II. USERS GUIDE

A. Getting started

To use the inventory control and billing program the following are needed:

1. A TRS-80 model III or 4 microcomputer, with at least 48K bytes of Random Access Memory (RAM) and two disk drives, capable of storing 170K bytes of information.
2. The program diskette and a formated blank diskette, (both 5 1/4 inch).
3. A printer 80 columns or larger.

B. Diskettes and disk drives

The diskettes are magnetically coated plastic disks inside a protective plastic cover. Through the oval cutout the magnetic surface is exposed. NEVER touch this area. Protect the diskettes from dust by storing them in the paper sleeve they came in. Keep diskettes away from magnetic fields (ie. TV). Extreme temperature can destroy diskettes. Don't bend diskettes or write on them with a hard pencil or pen, since the resulting indentation can physically damage the magnetic surface. Use a felt tip instead.

In the TRS-80 model III or 4, DRIVE 0 is the bottom drive and DRIVE 1 is the top drive. Whenever any drive is accessed, all drives turn and the red "active" light glows on the active drive. Never open the disk drive door to insert or remove a diskette while red lights are on, since this will probably damage the diskette and/or the drive. NEVER turn the power to your computer off or on while any diskettes are mounted.

C. Purpose

The INVENTORY CONTROL AND BILLING FOR A UNIVERSITY FEED MANUFACTURING FACILITY provides feedmill operators and departmental accountants with an efficient tool to account for transactions between the mill and the research units.

D. System capacities

15 Research Units within the Department

125 different ingredients in the mill inventory

20 ingredients maximimum per ration

E. General operating information

Before the program is used the first time, the program disks must be backed up or copied. You need four disks to begin, one is the original program disk and the three other are formatted blank disks (Apparat, 1981). Label the four disk as follows:

The orginial program disk labeled as - ORIGINAL DRIVE 0, SOURCE

Label disk two as - ORIGINAL DRIVE 1 , SOURCE

Label disk three - BACKUP, DRIVE 0, DESTINATION

Label disk four - BACKUP, DRIVE 1, DESTINATION

The original program disks are for daily use. The backup disks are used only if there is a problem with the original disks. Make new backups any time new information is entered. It only takes a few minutes and can prevent disaster later.

In this Users Guide, each time < > is seen this indicates you are to type in what is in the brackets followed by the <ENTER> key. <ENTER> means press the enter key.

Program setup starts with entering feed ingredients, their prices ($/lb) and amounts into the inventory list. When that is complete, rations can be entered. Since the program checks for the availability of ration ingredients, all ingredients used in a ration must be in the ingredient list. Once the initial setup is complete, purchases and sales can be recorded, new ingredients and rations added, and unused rations removed.

F. Problem solving procedure

The main menu is shown below, and is followed by a description of each entry.

```
K S U   F E E D M I L L
  M A I N   M E N U

1. ADD INGREDIENTS
2. EDIT,DELETE INGREDIENTS
3. ADD RATIONS
4. EDIT,DELETE RATIONS
5. REPORTS
6. DAILY SALES
7. DAILY PURCHASES
8. PURC & SALE CORRECTIONS
9. EXIT PROGRAM

ENTER NUMBER?
```

1. Add ingredients: new (not previously used) ingredients are added to the inventory list. An ingredient must be present on the inventory before it can be used in a ration.
2. Edit, delete ingredients: used only to adjust the inventory list. Use this in case of input error.
3. Add rations: used to add "new" rations to the ration file.
4. Delete, edit rations: used to adjust ration composition, but CANNOT increase the number of characters in a ration file.
5. Reports: printouts or hard copies of program output.
6. Daily ration sales: These are the ration sales to the individual units. An individual ingredient can be treated as a ration; for example, bulk corn to the swine unit.

7. Purchases: amounts and prices of purchased ingrdients are entered. Ingredient purchases are entered before sales are entered. Ration prices are based on the most current ingredient prices.
8. Sales and purchases correction: used to correct input errors in sales and purchases.
9. Exit program: executes a "backup" of important files, then ends the program.

G. The menu choices in more detail:

ADD INGREDIENTS

The "new" ingredients (those presently not in list) are added to the end of the ingredient file so that other ingredients are not disturbed. Each ingredient has a number, name, pounds in inventory and the purchase price per pound.

This section is used to initialize the ingredient file. The ingredient file must contain all ingredients that will be needed in rations before the ration file is initialized (item 3).

From the Main Menu enter selection 1 and <ENTER>. The program will automatically assign an incremented number to the new commodity. You supply the name, the amount of the commodity in inventory, and the purchase price (dollars per pound). The commodity name can be up to 15 characters long, including spaces.

The screen automatically shows the commodity number and asks for the commodity name. Enter the commodity name and <ENTER> when completed. The pounds of the commodity in inventory is added next, followed by the purchase price. Both pounds and initial value can have a zero value. This provides for commodities to be in inventory that were not previously used, as well as "no charge" items. Most ingredients will have initial values below $1.00/lb, if so enter a decimal value (ie. .50 for 50 cents). Enter up to five digits to the right of the decimal point. The complete screen looks like this.

```
                    ADD INGREDIENTS
        ENTER <E>DIT,<M>AIN,<ENTER> TO CONTINUE
**************************************************************

                    ENTER COMMODITY #:  104
                    ENTER COMMMODITY NAME: CORN
                    ENTER POUNDS: 2000
                    ENTER INITIAL PRICE($/LB): .0626
```

The next screen reviews the previously entered ingredient component and prints it at the top of the screen below the title.

```
                    ADD INGREDIENTS
        ENTER <E>DIT,<M>AIN,<ENTER> TO CONTINUE
**************************************************************
     NAME                NUMBER          QUANTITY          PRICE
     CORN                   104             2000          .0626
                             ? .
```

At the ? one of three options must be taken.

1. <ENTER> if data are correct and you wish to continue adding ingredients to inventory
2. <E> to edit data for the ingredient (explained more in next section)
3. <M> to return to the Main Menu.

By pressing the <ENTER> key at the ?, more ingredients may be added to the inventory list. The commodity number will be updated automatically.

The <E> or edit option allows changes or corrections in the ingredient on the screen. After pressing <E> the edit screen will be shown. The edit section explains in detail how to make changes.

When editing is complete, <ENTER> will return you to the "add ingredients" screen. Again the ? appears asking for the option.

<M> returns you directly to the Main Menu. There is one other way to return to the Main Menu; entering <M> when asked for the commodity name. This is a safety factor in case you accidently hit <ENTER> to continue.

EDIT, DELETE INGREDIENTS

The edit, delete ingredients option is to allow corrections of input errors. This time, choose option 2 from the Main Menu.

After the Main Menu choice has been made, the next question asked is:

```
ENTER COMMODITY <NUMBER> TO BE EDITED .
```

It is asking for the commodity number, not commodity name! You can print a list of commodity numbers and names by entering <H> from the Main Menu. This is explained further in the Help section. Enter the commodity number to be edited. The next screen looks like this:

```
COMMODITY NUMBER =  104
COMMODITY NAME = CORN
QUANTITY = 2000
VALUE ($/LB)= .0626

1. CHANGE ALL
2. CHANGE PRICES ONLY
3. CHANGE POUNDS ONLY
4. CHANGE NAME ONLY
5. DELETE ITEM FROM FILE
6. NO CHANGE FOR THIS DATA
ENTER YOUR CHOICE FROM THE OPTIONS ABOVE : .
```

It has located the commodity number and printed the information that goes along with it. From this menu you have six choices. Enter the number of your choice. Be careful with this menu, because once anything is changed, the calculations are then figured on these new values. Do not change the name unless you no longer have the ingredient in stock. Otherwise, when you sell rations, that ingredient will not be found. When all corrections are made select <6> to return to the Main Menu. Option 5 is used only infrequently to "clean up" files.

ADD RATIONS

To add a ration to the files, each commodity number required and pounds of each ingredient in the ration must be known. The rations will be divided up into the fifteen research unit files. The ration numbers are automatically incremented started at the first ration for a unit. Each ration contains a maximum of 20 ingredients .

From the Main Menu select <3>. The next screen looks like this:

```
                              UNITS
**************************************************************************
   1. HORSE UNIT                      9. DAIRY UNIT
   2. SHEEP UNIT                     10. POULTRY UNIT
   3. SWINE UNIT                     11. CASSODAY COWS
   4. PUREBRED UNIT                  12. KONZA COWS
   5. BEEF RESEARCH UNIT             13. BARTLEY'S COWS
   6. GRASS UNIT                     14. SPARE 1
   7. SHORTHORN UNIT                 15. SPARE 2
   8. COW-CALF UNIT

**************************************************************************
```

This is asking for the unit where the ration will used. Enter the appropriate unit number.

The next screen will first ask you for a Ration ID, up to 15 characters long. This can be ration name or an identification number, depending on the unit's method of ration identification. Enter the Ration ID and continue. At the bottom of the screen ingredient numbers (not names) are asked for. Enter the ingredient number. The next question is the number of pounds of this ingredient in the ration. The screen then will look like this:

```
RATION ADD, DELETE, EDIT        TOTAL LBS     200
RATION ID HORSE
NO.    ID#    INGREDIENT          LBS       PERCENT
 1      1   CORN                200.000       0.0000
```

```
INGRED. NO. ? 4                 POUNDS? 100
```

After all the ingredients in the ration have been entered, instead of entering a new ingredient at the INGRED. NO. prompt, enter an. <@>. This indicates that all ingredients have been added and the ration is complete. The screen looks like this:

```
RATION ADD, DELETE, EDIT        TOTAL LBS     320
RATION ID HORSE
NO.    ID#    INGREDIENT          LBS       PERCENT
 1       1   CORN                200.000      0.0000
 2       4   OATS                100.000      0.0000
 3       5   BRAN                 20.000      0.0000



INGRED. NO. ? @
```

Next, the computer sums the pounds of ingredients, computes the percentages, and reprints the screen:

```
RATION ADD, DELETE, EDIT        TOTAL LBS     320        100.000%
RATION ID HORSE
NO.    ID#    INGREDIENT          LBS       PERCENT
 1       1   CORN                200.000     62.5000
 2       4   OATS                100.000     31.2500
 3       5   BRAN                 20.000      6.2500



<C>ONTINUE,<S>AVE, <R>EVIEW, <E>DIT, RETURN TO <M>ENU?
```

There are five options at this point.

a. <C>ontinue to enter rations

b. <S>ave is the most important option. It saves the ration.

c. <R>eview allows the complete ration to be reviewed.

d. <E>dit the ration shown.

e. <M>enu returns to Main Menu.

The <C> allows you to continue entering rations without going back to the Main Menu. Only continue after the ration has already been saved. It goes to a continuation menu like the one below.

```
                    MORE RATIONS
1. SAME UNIT
2. DIFFERENT UNIT
3. MAIN MENU
ENTER NUMBER? .
```

From this menu, <1> means continue adding rations for the same unit, <2> means you want to continue adding rations, but for a different unit. Selection <3> return to the Main Menu.

The <S> from the ration addition menu is used to save the ration that you just entered. This is very important because If the ration is not saved it will never be acknowledged as being entered. IMPORTANT: if the ration is correct then immediately save it. This automatically saves the ration in the appropriate unit's file at the next available position in the file. However, do not save a ration twice because this only wastes space and gives the ration two numbers. If a ration is not correct, edit it before saving it.

To review the complete ration, enter <R>. This is especially useful when there are more then 10 ingredients in the ration. It will scroll through screen one (which only holds 10 ingredients) and then scroll through screen two to allow both screens to be seen.

To edit the ration enter <E>. Answer the appropriate questions. The first question is for which LINE (not commodity number) to edit. Next the commodity number and corrected pounds will be requested. The appropriate changes will be made in the ration. For further explanation of editing a ration, review the next section. The edit screen looks like this:

```
RATION ADD, DELETE, EDIT        TOTAL LBS     420        100.000%
RATION ID HORSE 2
NO.    ID#    INGREDIENT          LBS       PERCENT
 1      1    CORN                200.000     47.6190
 2      4    OATS                175.000     41.6667
 3      5    BRAN                 45.000     10.7143
```

```
EDIT NO. ? 2  INGRED. NO. ? 4      POUNDS ? 150.
```

The last selection available is <M> for Main Menu. This will go directly back to the Main Menu. If the ration has not been saved it will be lost.

EDIT, DELETE RATION

This section is used to correct input errors or ration changes. However, due to the way the rations are stored there is one stipulation; once a ration has been saved, you work in percents rather then pounds. Let's assume that a ration has four ingredients. The number of ingredients can not be increased. Also, once the number of characters in an ingredient field is established it can not be increased; for instance, ingredient number 4 was entered and

saved in the ration but it should of been number 40. The type of files being used will not allow the number of digits to increase, so to correct the problem, add a whole new ration.

When editing rations as they are being added (prior to <S>ave), see page 15 under <E> for instructions. When editing through the Edit Ration section, enter the number to be edited, then the corrected ingredient number, then the percent. The screen should resemble this.

```
RATION ADD, DELETE, EDIT        TOTAL LBS     395        100.000%
RATION ID S322B
NO.    ID#   INGREDIENT            LBS      PERCENT
 1      2    MILO                  0.000     63.2540
 2      3    SBM                   0.000     21.9956
 3      8    DICAL                 0.000      2.0996
 4      9    LIMESTONE             0.000      1.4064
 5     10    STOCK SALT  167       0.000      0.5666
 6     11    Z-10 TM               0.000      0.1133
 7     25    KSU SWINE PRMX.       0.000      0.5666
 8     24    ANIMAL FAT            0.000      9.9980

EDIT NO. ? 1  INGRED. NO. ? 1      PERCENT ? 63.254
```

IMPORTANT: after editing, <S>ave the ration. If no changes are made there is no need to resave the ration. The <R>eview command will review the ration. The continuation command will not be accepted in this case.

The edited ration must add up to between 101 and 99% or it will not be accepted. The range allows for rounding errors. This rounding error problem will only occur on those rations that have been edited.

REPORTS

The reports section (<5> from main menu) is designed to give the user a hard copy of day-to-day activities as well as month-end reports. All selections in the reports menu result in printouts. Examples of the different types of reports, are given in APPENDIX A. These should help to understand the individual selections. The Report menu is listed below.

REPORT MENU

****

1. MONTH END REPORT
2. INVENTORY LIST
3. RATION LIST
   (RATION CONTENT)
4. PURCHASES TO DATE(MONTH)
5. SALES TO DATE(MONTH)
6. RETURN TO MAIN

The first menu selection is Month-End report. This gives a complete breakdown of the ration sales. The month-end reports menu can be divided up into two main areas and one subarea. The subarea is a current inventory list at this point and time in the month, and is the same inventory list that can be obtained from option two of the reports menu. The two main areas are month-to-date reports and month-end close outs. The month-to-date reports are for use at any time in the month.

The individual ingredients within each ration sale are broken down and recorded by unit. For example, a listing of all the ingredients used by research unit number 5 is presented. Assume two batches of ration have been prepared. In one there are 100 pounds

of corn (plus other ingredients) and the other is 200 pounds of corn. Then the total amount of corn sold to unit number five is 300 pounds.

Also in the month-to-date reports is a listing of costs broken down by project numbers. The final month-to-date report is a listing broken down by ration. The month-to-date reports are all seperated out by research unit. Examples are listed in Appendix A.6, A.7, and A.8.

Total dollars and total pounds per year for the unit as well as totals for the month are recorded and printed.

The month-end closeouts, (<2> from this menu) are exactly like the month-to-day reports except they close out the purchases and ingredient files for the month. Thus, all the information for the month must be recorded and correct before closing out the month. A backup should be made before attempting to close out the month. Check in Other Hints section (page 25) for further suggestions. The month-end closeout section also saves the month's information for calculating the year-to-date information. This keeps all the totals for that month and adds it to the proceeding months.

The last option from this menu, <3> returns to the main menu. The menu is listed below.

```
                        MONTH END REPORTS
**************************************************************

     1. MONTH END REPORTS
     2. INGREDIENT LIST
     3. RETURN TO REPORT MENU



**************************************************************
     ENTER NUMBER?
```

If option one is selected then a check screen comes up to make sure you have backups of both disks. This is especially important now because if there is an electrical surge or lightning hits there are several files opened at one time. They could possibly be destroyed. The check screen below makes sure this is what you want to do.

```
HAVE YOU MADE A BACKUP DISK FOR THE DAY YET?  IF YOU HAVE NOT  *
 STEP ONE : GO BACK TO MAIN MENU * STEP TWO : HIT NUMBER 8 *STEP
 THREE : ACT AS IF YOU ARE DONE FOR THE DAY * STEP FOUR  : AFTER
 YOU HAVE BACKED UP BOTH DISKS  AND YOU RECEIVE A THE END PROMPT
 REBOOT THE MACHINE(HIT THE ORANGE BUTTON IN UPPER RIGHT HAND
CORNER OF KEY BOARD  *STEP FIVE  :  WHEN MAIN MENU REAPPERS GO
BACK TO REPORTS MENU AND CONTINUE AS BEFORE!!! (HIT <C> TO
CONTINUE,<M>FOR MAIN MENU? .
```

The next selection in the Reports Menu is a complete inventory list, including commodity number, commodity name, pounds in inventory, and price in dollars per pound. An example is given in Appendix A.1

The third selection is a complete ration listing, separated by unit. First, a unit must be selected. The program lists all the rations on file for that particular unit, including the ingredient numbers, ingredient names, and the percent of each ingredient in the ration. An example is given in Appendix A.2

The fourth selection is a purchase listing. It lists the month's ingredient purchases in the order they occurred (Appendix A.3).

The last selection returns you to the main menu.

DAILY RATION SALES

Selection number six from the Main Menu is daily ration sales to the research units, and includes sales of individual ingredients. A completed ration sale to unit number 10 is listed below:

```
1. DATE (MO/DA/YR) - 07/01/83
2. UNIT# - 10
3. TYPE(<R>ATION OR <I>NGREDIENT) - R
4. RATION OR INGREDIENT # - 3
5. QUANTITY - 1000
6. PROJECT #  (OPTIONAL) -
7. <B>ULK OR <S>ACK - B
```

The first entry is the date. This can either be the date of the sale or the date the information was entered. The program remembers the previous date, so if several sales are made on the same date, use <Y> to use the last date for the next sale. If the date needs changed, just enter the corrected date and continue with the rest of the sale.

The second entry is the unit number to which the sale was made.

The third entry is the type of sale. For a ration sale use <R> and for an individual or bulk ingredient sale use <I>.

The ration or ingredient number is entered next. The ration number corresponds to the assigned ration number within that particular unit. The ingredient number, however, is from the inventory list. Next enter the total quanity of the sale in pounds.

The sixth item is the project number. This is optional but may be useful to the deparmental accountants to evaluate the project costs. <ENTER> if no project number is to be entered. Last, indicate

The sixth item is the project number. This is optional but may be useful to the deparmental accountants to evaluate the project costs. <ENTER> if no project number is to be entered. Last, indicate whether the sale is <B>ulk or <S>ack. Bulk rations are marked up 12% and while sack sales are marked up 18%.

Check the data. If correct, <Y> will save the sale and <N> will allow you to correct the information.

PURCHASES

Purchases must be entered before for sales for a paticular time period so that sales will reflect current ingredient prices. A completed purchase screen is listed below.

```
                              PURCHASES
**************************************************************************

     1.  DATE (MO/DA/YR) - 07/01/83
     2.  INGREDIENT # - 1     CORN
     3.  QUANTITY - 1000
     4.  PRICE ($/LB)  - .0626.
```

The date can be the date of purchase or the date the purchase was entered. The date is handled like the sales date by allowing an initial entry, then repeating it with <Y> if appropriate. The ingredient number is from the inventory list. For an ingredient to be purchased, the number must already be in the inventory list. Otherwise use the "Add Ingrdients" option from the Main Menu. The quanity is the pounds of the ingredient purchased. The price is the ingredient cost in dollars per pound. The program expects values

below one dollar per pound. If the price is higher, such as for antibiotics, the user is asked to verify the information. If all information is correct, then <Y> updates the inventory amounts, value, and price per pound. If the information is not correct, <N> enters the correction routine.

SALES AND PURCHASE CORRECTIONS

This selection is a backup in case the purchase and sales input errors were not noticed at entry. On the report sheets, purchases and sales are numbered. The correction menu will ask whether a purchase or sale correction is required. Both have a correction and a price adjustment. The correction is simply correcting an input error. The price adjustment for purchases, however, adjusts the price in case a special purchase causes the computed price to not correspond to the inventory list price.

```
                MENU FOR CORRECTIONS AND ADJUSTMENTS
**********************************************************************
                    1. PURCHASE CORRECTIONS
                    2. PURCHASE PRICE ADJUSTMENTS
                    3. SALES CORRECTION
                    4. MAIN MENU

                     ENTER SELECTIONS? .
```

Select an option from the menu. The top three selections ask for the sale or purchase number to be corrected, from the reports. Here is an example of a sale being corrected.

```
                SALES CORRECTIONS & ADJUSTMENTS
************************************************************
   WHICH SALE NEEDS CORRECTED OR ADJUSTED? 3
         1. DATE - 06/30/83
         2. UNIT #02
         3. TYPE (R OR I) - I
         4. ID# - 13
         5. QUANITY - 10
         6. PROJECT # -
         7. <B>ULK OR <S>ACK - B
         8. TOTAL PRICE - .729
         9. PRICE/LB$ -  .0729
 ARE THESE CORRECT(Y OR N)?? .
```

Just enter the number of the sale to be corrected and the program does the rest.

Purchase corrections work just like the sales corrections.

If price adjustments in purchases are to be made, just change the total value or dollars per pound, depending on the adjustment, and this value will be saved. These price adjustments are in special cases only. For instance, if a purchase is made at a certain dollars per pound, but the salesman donates the university extra grain, the program allows the price to be manually adjusted.

EXIT PROGRAM

This option executes a "backup" of all the files on the original disks. Follow these instructions closely. Be sure when you first started that you had the disks labeled correctly; ORIGINAL and BACKUP disks 0 (bottom drive) and ORIGINAL and BACKUP disk 1 (top drive).

The screen will say "ENTER BACKUP DISK 0 IN TOP DRIVE (DRIVE #1). HIT <ENTER> WHEN COMPLETE OR <M> FOR MAIN MENU". Assuming you wish to continue with the backup, make the appropriate disk mount and <ENTER>. The screen looks says:

STARTING DISKETTE COPY

FORMAT DISKETTE ? (Y OR N)<N>

ARE SYSTEM AND SOURCE THE SAME DISKETTE?(Y ORN)<Y>

PRESS ENTER WHEN DESTINATION DISKETTE MOUNTED ON DRIVE 1

<ENTER>

The active lights will indicate that the disk is being copied. When the backup is complete a second question is ask. Again follow the directions very closely.

The second prompt is "MOUNT BACKUP DISK #1 IN TOP DRIVE (DRIVE #1). HIT ENTER TO CONTINUE (FOLLOW DIRECTIONS THAT FOLLOW)". Make sure the disk in drive 1 is BACKUP DISK #1, not the ORIGINAL disk. Next the same questions will be asks as above but (IMPORTANT) these questions are answered differently.

STARTING DISKETTE COPY

FORMAT DISKETTE? (Y OR N) <N>

ARE SYSTEM AND SOURCE THE SAME DISKETTE(Y OR N)<N>

ENTER SOURCE DISKETTE IN DRIVE 0 AND <ENTER>

PRESS ENTER WHEN DESTINATION DISKETTE MOUNTED ON

DRIVE 1 <ENTER>

The original disk #1 is not a system disk so the third question is answered <NO>. When you copied the origial disk 0 to backup 0, the diskette in drive 0 was a system diskette. NOTE!! THE DIFFERENCE in answers to the questions. When the source diskette is ask for in this last case, it is main disk #1, mounted on drive 0,

after this is completed there is an indication of the end of program. Remove the disks, then turn off the printer and the computer.

HELP ROUTINE

The help routine is accessed from the main menu by selecting <H>. The help routine offers some additional information that only needs to be accessed once in a while. The help menu is listed below.

```
                        HELP ROUTINE
**************************************************************
          1. INGREDIENT LIST
          2. INDIVIDUAL INGREDIENT INFORMATION
          3.  MAIN MENU
          4.  RATION LIST NAMES
          5.  ALPHABETIZED INGREDIENT LIST



ENTER NUMBER
?
```

Selection <1> sends to the screen a complete list of the ingredients in inventory. Selection <2> seperates out an individaul ingredient. Selection <3> returns you back to the Main Menu. Selection <4> gives you a hard copy of the ration ID and ration number for a particular unit (Appendix A.5). Selection <5> gives an alphabetized inventory list (Appendix A.9).

OTHER HINTS

It is recommended that you use two sets of backups, and alternate their use between days. This is a safety factor in case the information entered that day is somehow lost.

If the *Feedmill* program indicates an error, it is probably a disk operating system error. Most other errors have been trapped by the program (Apparat, 1981).

III. TECHNICAL GUIDE

A. Introduction to Technical Guide

This feedmill management system is made up of two separate programs, one for purchase and sales operations (Main program) and one for reports. Each will be discussed separately.

The Technical Guide includes information on file types and file methods used as well as an inside look at the program. Major components are identified and the purpose of each is noted.

Source code for the system, is in Appendix B. However, to save RAM space, the user version has had all remarks and spaces removed, using CMD"C" in NEWDOS/80.2. Any modifications should be made on a copy of the source version, comments added, then compression repeated.

B. File type definitions. (Apparat, 1981)

FF A randomly accessed file with fixed and equal record lengths. Record length and file type are specified in the OPEN statement.

MU A random access file segmented into records of varying lengths, where the length is delimited by the next record's SOR (start of record), or the EOF (end of file) markers. Length is optional, but if specified, is used as the maximum allowable record length. A MU file may be updated provided the record length is not increased beyond its original value. If the record is shortened, the empty space is occupied by fill items.

C. Special features

Error Functions

ON ERROR GOTO 65000 If error then goes to Main Menu

Other functions and statements: (Tandy, 1980B)

POKE, 16425,0	Sets the printer line counter to 0
POKE, 15424,67	Sets lines per page to 66
POKE, 16916,X	The top X lines of the screen don't scroll
POKE, 16916,0	No screen lines are scroll-protected
CHR$(23)	Double-size characters
CHR$(12)	Top of form command.
CHR$(30)	Erase to end of screen line
CHR$(31)	Erase to end of display
PEEK (16425)	Printer lines since last top of form
PEEK (14312)	Printer status check

MAIN PROGRAM

D. File information

File name	File type	Record length	Comment
INV	FF	36	Inventory list
SALE:1	FF	51	Individual sales
PURC:1	FF	50	Individual purchases
ING:1	FF	44	All individual ingredients sold
MU/SH	MU	300	Shorthorn unit rations
MU/BF	MU	300	Beef unit rations
MU/SW	MU	300	Swine unit rations
MU/GR	MU	300	Grass unit rations
MU/SP	MU	300	Sheep unit rations
MU/PB	MU	300	Purebred unit raitons
MU/H	MU	300	Horse unit rations
MU/PT	MU	300	Poultry unit rations
MU/COW	MU	300	Cow-calf unit rations

MU/CA	MU	200	Cassady cow ration
MU/KON	MU	200	Konza rations
MU/BT	MU	200	Fistulated cow rations
MU/D	MU	200	Dairy rations
MU/SPA	MU	200	Spare one
MU/SPB	MU	200	Spare two
MU/SPC	MU	200	Spare three

E. Major components (routines or subroutines)

1. Main Menu routine
2. Backup routine
3. Add ingredients routine
4. Inventory file editor routine
5. Ration addition and editor routine
6. Sales routine
7. Individual ingredient sales routine
8. Ration sales routine
9. Help routine
10. Unit listing report subroutine
11. Purchases routine
12. Individual ingredients sales routine
13. Month-end reports subroutine
14. Report menu routine
15. Ration unit list subroutine
16. Inventory list subroutine
17. Purchases reports subroutine
18. Sales reports subroutine

19. Error subroutine
20. Yes/No subroutine
21. Date error subroutine
22. Feedmill picture subroutine
23. Sales correction subroutine
24. Ingredient list subroutine
25. Individual Ingredient list subroutine
26. Purchase correction subroutine
27. Correction menu routine
28. Inventory check subroutine
29. Alphabetized inventory list

F. Variables and Arrays:

This is a complete list of the arrays and variables used in the Main Program. Included are comments and a list of the routines in which they occur. All numeric keyboard entries are checked through the error subroutines for varification. Numeric values are entered as character strings then either saved as character strings or converted to numeric values for computation. The variables below are not designated string or numeric because they all start out as strings and if need be are changed to numeric.

Item group expression list (IGEL) is a term used by NEWDOS/80.2 files to designate the number of spaces assigned to variables. It is similar to the FIELD statement used in other disk operating systems. Below are the IGEL's used, and the files where they are used. The file name that corresponds with the group of file items is also noted.

INV	(5)CN$,(15)N$,(10)LB$,(6)P$;
SALE:1	(10)D1$,(2)U1$,(1)G$,(5)B1$,(10)Q1$,(15)PJ$,(2)S1$,(6)P1$;
PURC:1	(8)D2$,(5)I1$,(10)Q2$,(6)V1$,(10)P3$;
ING:1	(5)U1$,(6)CN$,(12)Q$,(15)N$,(6)P2$;
All MU's	Z1$;

Array Variables

Array &	Dim	Comment	Routine (page 29)
DA	20	Ingredient #	1,8,12
DB	20	Ingredient name	1,8
DC	20	Pounds of ingredient	1,8
DE	20	Price of ingredient	1,8
DF	20	Quanity	1,8

IN	20	Ration ingredient list	1,5,8,15,23
IV	125	Ingredient names	1,5,15,17,22
PC	20	Percent of ingredient in ration	5,8,15
WT	20	pounds of an ingredient in ration	5

Non-array variables

Variable	Comment	Routine (page 29)
A	Ration addition flag	1,5,8,10,15
A	Input value	2,4,5,7,8,13,26,27
A	Print format	5
A1$	String of *	3
A1$	Input value	23
AA	Ration addition flag	1,5,26
AV$	Sales report "print using" format	18
AX$	Purchase report "print using" format	17
AZ$	Unit listing "print using" format	15
B	Ration number	5,6
B	Input value	23
B1	Ration number	5
B1	Ingredient number	7
B1	Ration or ingredient number	6,23
B2	Input value	1
B5	Ration or ingredient number	23
BB	Ration or ingredient number	23
BY	Input value	14
C	Remaining ration record minus RI	5

C2	Input value	11
CA	Ration Id	5
CB	Position location in ration	5
CC	Position location in ration	5
CJ	Length of Q2	5
CN	Commodity number	3,4,23,25
CO	Counter for Ration Id length	5
CR	Counter for purchases	11
D	Indicator start of ingred. in record	5
D	Counter for ingredients	24
D1	Date	6,18,23
D2	Date	11,17,26
DE	Total pounds of purchases	17
DE	Total pounds sold	18
DF	Total cost of purchases	17
E	input value	5
ED	Flag for corrections	27
ET	Input value	14
F	Counter for screen position	24
F9	Unit list number counter	10,15
FD	Flag for correction	26,27
FG	Flag for sales correction	8,23
G	Ration or ingredient sale type	6,12,18,23
G5	Ration or Ingredient sale type	23
GG	Ration or Ingredient sale type	23
H	Putting ingredient together(record)	5

I	Loop indicator	5,8,23
I	Counter	5,12,16,17,22,24
II	Ingredient number	11,117,26
IA	Input value for sale number	23
IN	Ingredient number	5
J	Input value	6
J	Loop indicator	19,21,22
J1	Input value	6
J2	Input value	6
JJ	Inventory format	16
K	Loop indicator	5
L	Input value	5
L	Counter	24
LB	Pounds of an ingredient	3,4,7,8,10,11,16,
LB		24,26
M	Input value	3,4
M	Ration edit flag	5
M	Unit number	5,6,8,23
N	Commodity name	3,4,7,8,16,22,24,25
N	Ingredient counter	5
N	Loop indicator	5,15
N1	Page ingredient counter	5
N1	Ingredient name	24
N2	Ration ingredient counter	5,8,15
NA	Ingredient name	5,15
O	Input value	5,9,24,25

P	Ingredient price	3,4,7,8,11,16,25,26
P1	Total ratio sale price/lb	6,7,8,18,23
P2	$/lb of each ingred in ration	7,8
P3	Total price of purchase	11,17,26
P3	Price of sale	23
P4	Total price of sale	23
PJ	Project number or name	6,18,23
PO	Enoungh lbs of ingredient flag	6,7,8
PP	Purchase flag	11
PP	Sale $/lb	23
Q	Length of QZ minus one	5
Q	Quanity of Ingred. sale	7,8
Q1	Input quanity of sale	6,7,8,18,23
Q2	Purchase quanity	11,17,26
Q5	Correct sale quanity	23
QQ	Temp. sales quanity	23
QZ	Remaining ration record	5
R	Input value (inventory editor)	4
R	Setting R to 0	3
RI	Ration ID name	5,10,15
S1	Bulk or Sack	6,7,8,18,23
S2	Correction Bulk or Sack	23
S5	Temp. Bulk or Sack	23
ST	Printer status	14
T	Counter for complete ingredient	5
T1	Inkey value	6,11

TL	Total pounds in ration	5
TP	Total percent in ration	5
U1	Unit number	6,7,12,18,23
U5	Temp unit number	23
UU	Temp unit number	23
V1	Purchase $/lb	11,16,26
W	Edit, delete ration flag	1,5,8
WW	Edit, delete raition flag	1,5,8,10,15
X	Commodity number	12
X9	INV's LOF (last of file record number)	1,4,5
X9	Commodity number	3
Y1	yes/no sub coordinator	3,4,20,23
Y5	Input value	20
YY	Ration addition idicator	1,5
Z	Error & Date coordinators	1,3,4,5,6,10,11,13
		19,21,23,26
Z1	Completed ration (record)	5
Z9	Input value	3
ZQ	Total cost sale	18
ZX	Continuation input value	5
ZZ	Value of purchase	17
ZZ	Error position by position check	19,21

G. Description of individual program components. The numbers are line numbers of the source code (Appedix B.1)

1. MAIN MENU

Purpose: routine prints the main menu to the screen, then calls other routines as selected.

1 - 17

Dimensions the arrays and puts the Feedmill picture on the screen.

20 - 172

Prints the main menu to the screen and asks for a selection (B2$).

180 - 210

Depending on the value of the selection other routines are accessed. If B2$ is "H" then the Help routine is called. If B2$ is "3" or "4" then flags A$, YY, AA$, W, and WW are set.

2. BACKUP ROUTINE

Purpose: prepares a backup of the original disk.

220 - 360

The screen displays instructions for, making sure the proper disks are mounted. The DOS copy command, CMD"COPY 0 1" copies the disk in drive 0 to the disk in drive 1. For disk instructions consult the Users guide.

3. ADD INGREDIENTS

Purpose: adds new ingredients to the inventory list. This routine is accessed by the main menu selection <1>. It creates and updates the file "INV".

500 - 635

The file "INV" is opened. The commodity number is incremented and printed. The variable X9 (commodity number counter) is initialized in the Feedmill Picture routine. The other information for inventory is entered. The commodity name can be up to 15 characters long. The initial price of the commodity is in dollars per pound. If the price is greater than $1.00 (antibotics for example) then the operator is asked to verify.

640 - 770

At the top of the screen there is a menu. The information just entered is printed below this is to confirm the input. The menu consists of <E>dit, <M>ain, or <ENTER> to continue. To clarify what each does consult the users guide.

4. EDIT, DELETE INVENTORY

Purpose: edits records in the "INV" file. It is only used to correct input errors.

780 - 1131

The file "INV" is opened. The commodity number to be edited is entered. If the commodity number is too large it will not be accepted. The commodity number is equal to the record number in "INV". This record number is accessed and the information associated with it printed to the screen along with a menu. Make the changes according to the menu. After the information is correct select <6>. This returns you back to where the routine was called. The two places of origin for this subroutine are the Main Menu or the Add Ingredients Routine.

5. RATION ADDITION

Pupose: handles all the information in ration (MU) files.

3000 - 3169

The unit names and their associated unit numbers are printed on the screen. A unit number is selected, and the appropriate files are opened in 3170-3495.

3170 - 3495

A is a flag. If A = 1, there are more rations to be added, and the files are prepared for extension. Otherwise, the ration files are opened randomly (Ration Edit). If new rations are to be added, control passes to line 2000. If the rations are to be edited, control passes to line 3500.

2000 - 2150

The ration is entered in this section. First, the Ration Id is entered, followed by entering the ingredient number and pounds in the ration. The ingredient numbers and pounds are alternated until the complete ration is entered. After the ration is completely entered then <ENTER> and <@>. This indicates the ration is complete. The percentages are figured and printed on the screen. A menu also appears with five options: <S>ave, <R>eview, <M>ain, <C>ontinue, or <E>dit.

2190 - 2270

If the menu selection is <S>, ration data are saved in one long character string. First, Z1$ is the concatination of ingredients and percents, alternating between them. The ration ID and a flag "Z" are

appended to the beginning of Z1$. The new Z1$ is stored in the next available position of the unit's MU file.

2280 - 2300

If the menu selection is <E> then the ration is ready to be edited. An edit number is ask for. This is the edit line number, not ingredient number. The ingredient number and pounds or percent are then entered. If the ration has not been saved, enter pounds. Once the ration has been saved, enter the corrected percent. Once this information has been entered the corrected screen will be shown.

2310

This subroutine within Ration Addition simply prints the ingredient and pound information in the proper spot on the screen after entry.

2320 - 2340

This subroutine prints the titles and ration ID in the proper places.

2400 -2430

This prints the continuation menu with a possibility of three selections. Selection <1> (same unit) goes to line number 2000. Selection <2> (different unit) goes to 3000 to allow another unit to be chosen.

3500 - 3710

If both flags W and WW are 0 then the operator is asked for the ration number to be edited. If W and WW are not equal to 0 the program already knows the ration number. The ration number is assigned to variable B. The file for this particular unit has already been opened in lines 3400 - 3495. The record number is obtained by a loop, starting with the first record in the file and continuing until

the record is equal to the ration number. This record information is assigned to Z1$. This variable contains the complete ration information. To seperate the components in this ration, each character is checked. First "Z" is found. All characters proceeding it are the ration ID. This is done by the MID$ command one poition at a time until a Z is found followed by a space. The ration ID is assigned to CA$. Next the ingredients and percents have to be seperated. C$ is equal to the remaining record after the ration Id and Z flag are subtracted. The ingredients and percents alternate in the ration, so first an ingredient must be found. The string is examined position by position using the MID$ command. A space indicates the end of an ingredient and the start of a percent. The percent is reviewed in same manner. As the ingredients and percents are found they are put into the arrays IN$ and PC$. When the ration is complete it is returned to the ration sale component and the appropriate calculation carried out. For editing, control passes to line number 2270 to allow the appropriate changes.

6. DAILY SALES ROUTINE

Purpose: handles the daily sale of rations and ingredients to the research units.

4000 - 4200

The date (either the date of the sale or the date of entry) is entered, in the MO/DA/YR format. It will be saved and can be used until it needs to be changed by entering <Y> when the date query appears (line 4027-4028). If the date needs changed, type over the old date. Data entered are unit number, type (ration or ingredient), ration or

ingredient number, quantity, project number, and bulk or sack. The operator is asked for verification; "Are these correct". If correct, enter <Y>; if incorrrect, <N>, (this is an INKEY$ function). If incorrect, correct the data by the steps in next section. G$ is "R" or "I" indicating ration or ingredient. If G$="R" this routine calls the subroutine for ration sales. If G$="I" then the routine calls ingredient sales.

4201 - 4218

If the sale information is incorrect, correct the information and continue.

4219 - 4280

The sale file is opened and the information is stored in the next available record position (line number 4225). After the sale has been stored two options are presented; either continue or go back to the main menu.

7. INGREDIENT SALE SUBROUTINE

Purpose: to figure the cost of a individual ingredient sale.

5000 - 5060

"INV" is opened in buffer 1 and "ING" is opened in buffer 2. The ingredient number is assigned to B1, and is equal to the record number. The information in the inventory list is accessed. The value of the sale is computed in 5040, by multiplying the quantity of the sale (in lbs) by the price ($/lb) in inventory. This product is multiplied by 1.18 (markup) if it is a sack sale or by 1.12 if bulk. The pounds in inventory are reduced by the quantity of the sale, and the data

stored back into the inventory file. Next, the Individual Ingredients subroutine is called.

8. RATION SALE SUBROUTINE

Purpose: computes the value of ration sales.

5500 - 5520

The ration sale subroutine first accesses the ration routine to determine the ration contents. The flag WW=2 informs the ration routine that this is a sale.

5525 - 5610

This section resembles the Ingredient Sale routine. The difference is that the quantity used of each ingredient must be figured from multiplying the quantity of ration made by its percent in the ration (line 5560). Reference the ingredient sale subroutine for pricing calculations. Also if the flag FG=1, a sale is being zeroed out or corrected, so quantities are to be added back into inventory. This is done by subtracting the quantity from zero, to yield a negative value. Then in line 5580 when the quanity of the sale is subtracted from that in inventory, the inventory inceases. In either case each ingredient in the ration is put in an array. This makes sure the sale is possible.

5620 - 5651

The information is saved back into the inventory file ingredient by ingredient.

9. HELP ROUTINE

Purpose: to assist the user during program operation.

6000 - 6200

Various options are called from the HELP menu.

10. UNIT LISTING SUBROUTINE

Pupose: generates a list of rations, by unit.

6300 - 6610

This subroutine gives a hard copy of the ration names. It accesses the specified unit files, and calls and prints the ration numbers and their names, separated by unit.

11. PURCHASES

Purpose: to record the purchases of ingredients already present in the inventory list.

7000 - 7103

The date is entered just like the sales date, using <Y> to use the previous date. The ingredient number, quanity, and price ($/lb) are entered.

7110 - 7270

The "INV" file is opened, the record for the proper ingredient accessed, and the quanity of the purchase added to the quanity already in inventory. An updated inventory price is computed in 7140, as follows: The value of the original inventory is computed by multiplying quantity in inventory by its unit price. The value of the new purchase is added to that figure to yield the total value of the ingredient in inventory. The total quantity in inventory is computed by adding the original inventory and the new purchase. The the new price is computed by dividing total value by total quantity in inventory. The new inventory information is replaced back in th

"INV" file. The file "PURC:1" is opened in 7200. The total value of the purchase is computed and the purchase information is stored in the "PURC:1" file. Either <ENTER> to continue or go back to main menu (<M>).

12. INGREDIENT SUBROUTINE

Purpose: saves the information associated with each individual ration or ingredient sale.

8400 - 8440

This subroutine is called from both the Ingredient Sale and the Ration Sale routines. Numbers less then ten are padded with leading zeros, and the information is stored in the "ING" file.

13. MONTH-END REPORTS

Pupose: generates month-end reports. It is called from the Reports menu (selection <1>).

8800 - 8980

The menu is printed on the screen. The first option is the month-end reports. This option loads and runs the "REPORTS" program. Selection <2> calls the same ingredient subroutine as selection <2> from the reports menu. The third selection returns control to the main menu.

14. REPORTS MENU ROUTINE

Purpose: to print the reports menu and call the various subroutines.

10000 - 10170

After the reports menu is printed it checks the status of the printer (10140 -10144). Then the subroutines are called to give the desired printouts.

15. UNIT LISTING

Purpose: generates a complete ration list for the unit specified.

12100 - 12250

Prints out all the rations and their composition, arranged by unit. The subroutine calls the ration routine and gets the proper information.

16. INVENTORY REPORT

Purpose: to generate a current list of quantities of inventory ingredients and their prices.

13000 - 15375

The INV file is opened and accessed record by record, and data printed until EOF is reached.

17. PURCHASE REPORT

Purpose: to print a record of purchases made to date in a month.

16000 - 16120

The "PURC" file is opened, the records are accessed in the order of entry, and records printed. The quantities entered, and the costs of the purchases are each summed. Totals are printed at the end of the report.

18. SALES TO DATE REPORTS

Purpose: to genterate a report of sales-to-date within a month.

17000 - 17100

The "SALE" file is opened and each record printed until EOF is reached.

19. ERROR SUBROUTINE

Purpose: verifies that entries are numeric. This routine is accessed each time a numeric value is entered form the keyboard.

40000 - 40100

The entry string is checked position by position with a MID$ command, and asks that the entry be repeated if it detects any charcters other than 0-9 and certain characters such as ".", "@", and "-".

20. YES - NO SUBROUTINE

Purpose: to verify that yes/no questions are answered as such.

40300 - 40340

This subroutine is called each time yes/no questions are being ask. If a response other than <Y> or <N> is given, and error is indicated and another response requested.

21. DATE ERROR SUBROUTINE

Purpose: verifies that date entries are within a valid range

45000 - 45155

Year must be greater than 82, the number of days less than 32 and the months number less than 13.

22. FEEDMILL PICTURE ROUTINE

Purpose: to print a picture of a feedmil on the screen and load the inventory names in to an array.

50000 - 50170

The feedmill picture is composed of graphics characters. The "INV" file is opened and the ingredient names are read into the array IV$.

This array is used throughout the program to print ingredient names with the ingredient number.

23. SALES CORRECTION SUBROUTINE

Purpose: to conect sales data with input errors not caught when entered.

This subroutine is called from the corrections menu.

51000 - 51301

Sales transactions are numbered in the order they were entered. Line 51040 asks for the sales transaction to be corrected. The "SALE" file is opened and the record number (sale number) is accessed. The information is printed on the screen. The number within the list that needs changed is asked for (51180). The corrected information is entered (lines 51220 -51300).

51305 - 51480

If the value A1$ is "1" or "6" then the information is simply changed and restored. If it is "8" a corrected total cost and a correct price ($/lb) are figured, and the results restored. If it is "9" this new value is stored. If it is "7" then the mark up value is correctd and restored. Otherwise, "SALE" must be corrected by recomputing. First, the sale is zeroed out by assigning all values a negative value (this is done in ration and ingredient sale subroutines). The "ING" and "INV" file are recalculated, subtracting the sale from "ING" and adding it back to the "INV" file. The effect of the sale has been removed from the files. After this is done the corrections are made, and the new sale is recomputed and resaved over the old sales record.

24. INGREDIENT LIST SUBROUTINE

Purpose: prints to the screen a complete list of ingredient names in inventory. This subroutine is called from the HELP routine.

60030 - 60280

The "INV" file is opened and the inventory list is accessed. Record numbers and ingredient names are printed on the screen forty-two at a time until it has listed all inventory items.

25. INDIVIDUAL INGREDIENT SUBROUTINE

Purpose: prints to the screen an individual record out of inventory. This subroutine is called from the Help routine.

60300 - 60410

The "INV" file is opened and the complete information about each ingredient (record) is printed to the screen.

26. PURCHASE CORRECTIONS SUBROUTINE

Purpose: to correct the purchase file in case of input errors.

61000 - 61310

This subroutine is called from the corrections menu. Each purchase transaction had been assigned a number, which is also the record number. This value is the record number that is in the "PURC" file. The information in the record is printed to the screen. If the purchase is not correct, it is subtracted from inventory. After the corrections are made the purchase is recalculated and stored over the old purchase, and the inventory adjusted.

27. SALES AND PURCHASES CORRECTION ROUTINE

Purpose: prints the corrections menu on the screen.

62000 - 62140

This routine is called from the main menu. It prints the correction menu and calls the two corrections subroutines.

28. INVENTORY CHECK SUBROUTINE

63000-63100

This routine is called from the Add Ingredients subroutine. It searches the inventory list for ingredients to avoid duplication in inventory.

29. ALPHABETIZED INVENTORY

63500-63600

An alpahbetized inventory list is sent to the printer. This routine is called from the Help subroutine. It performs an indirect sort (using CMD"O") on the array IV$ and prints the results.

65000

If an input error exsits the program will RESUME at the main menu.

MONTH-END REPORTS PROGRAM

The month-end reports program allows a total month's summary to be printed. It is divided into month-to-date reports and month-end closeouts. The printouts look identical but the month-end close report is printed out and files for the month are either killed or closed out in preparation for starting the next month. There are three types of reports. The first is an individual ingredient list, by unit, of all ingredients and the quanity used for the month. This breaks down all rations by ingredient and treats them as individual ingredients. The second report is broken down by project numbers within each unit. The third is a ration use list by ingredients. An example of each can be found in Appendix A.

This program is accessed from the main program through the reports menu (<1>). It wants to know how many copies of this report to generate, and whether it is to be a month-end closeout or just a month-to-date report.

H. Special Features

CMD"O" - The built in BASIC array indirect sort. An index array is set up to access the other arrays.

CMD"F=KEEP" - causes all variables to be cleared except those specified.

CMD"F=ERASE" - allows the specified variables to be cleared.

I. File information

File name	File type	Record Length	Comment
ING:1	FF	44	Months ingredient sales
TOT	FF	42	Years ingredient use
SALE:1	FF	51	Months sales
TPROJ	FF	36	Years project sale totals
TRAT	FF	40	Years ration sale totals
P	FF	20	Purchases total

J. Major routines

1. Month-end ingredient reports
2. Month-end project reports
3. Month-end ration reports
4. Month-end closeout

K. Variable and Arrays

IGEL expressions

ING (5)U1$,(6)CN$,(12)Q$,(15)N$,(6)P2$;

TOT (5)U1$,(15)N$,(6)CN$,(9)TG$,(7)PT$;

SALES (10)D1$,(2)U1$,(1)G$,(5)B1$,(10)Q1$,(15)PJ$,(2)S1$,(6)P1$;

TPROJ (3)U1$,(3)PJ$,(10)QQ$,(8)P5$;

TRAT (2)U1$,(3)RN$,(15)PJ$,(10)QQ$,(8)PP$;

P (10)DF$,(10)DE$;

Array Variables

Arrays & Dim		Comment
AK	300	Sorted ingredient array unit number
CK	300	" " commodity number
DK	300	" " quanity
EK	300	" " price of sale
FK	300	" " pounds for year
GK	300	" " value for year
AS	250	Unit number in TRAT
BS	250	Ration number in TRAT
CS	250	Project number in TRAT
DS	250	Quanity in TRAT
ES	250	Price in TRAT

F1	250	Unit number final project list for month
F2	250	Project number final project list
F3	250	Quanity for month project list
F4	250	$/lb Price for month project list
F5	250	Total value for month project list
F6	250	Quanity for year project list
F7	250	$/year value project list
K1	550	Unit number in ING file
K3	550	Commodity number in ING file
K4	550	Quanity in ING file
K5	550	Price in ING file
KA	300	Unit number in TOT file
KC	300	Commodity number in TOT file
KD	300	Quanity in TOT file
KE	300	Price in the TOT file
MA	250	Unit number for ration list
MB	250	Ration number for ration list
MC	250	Quanity for month ration list
MD	250	Price ($/lb) for ration list
ME	250	Quanity for year ration list
MF	250	Total $ year for ration list
S1	150	Unit number in SALE
S2	150	Project number in SALE
S3	150	Quanity in SALE
S4	150	Value in SALE
S5	150	Ration or Ingredient in SALE

S6	150	Ration or Ingredient number SALE
SA	250	Unit number in TPROJ
SB	250	Project number in TPROJ
SC	250	Quanity in TPROJ
SD	250	Value in TPROJ
X1	250	Unit number final ingred. list for month
X3	250	Commodity number final ingred. list
X4	250	Quanity final ingred. listings
X5	250	Value final ingred. listing
A	550	ING index file
B	300	TOT index file
C	150	SALE index file
D	250	TPROJ indes file
E	250	TRAT index file

Non-array Variables

Variable	Comment	Routine (page 52)
A	Input value	1,2
AI	Counter for A% index file	1
AJ	Counter for B% index file	1
AL	Counter for E% index file	3
AM	Counter for C% index file	3
AN	Counter for E% indexfile	3
AT	Counter for TRAT file	2
BI	Ration or ingredient #	2
BT	Counter for sale array	2
CI	Position of sale index files	3

CC	Position of sale index file	3
CN	Commodity number	1,2,3
CO	Counter in ingredients printount	1
CT	Counter in ING list	1
DD	Position counter in D%	3
DF	Total purchase format	4
E1	Positon counter for E%	2,3
G	Type ration or ingredient	2
H1	Total month pounds /unit	1
H2	Total month price /unit	1
H3	Total year pounds /unit	1
H4	Total year price /unit	1
H5	Total month pounds total	1
H6	Total month price total	1
H7	Total year pounds total	1
H8	Total year price total	1
II	Counter for MA% array	2
J	Loop indicator	1
JJ	Array counter for AK%	1
KK	Month-end closeout flag	1,3
LL	Counter for array F7	3
MM	Flag for totals	1,2,3
N	Ingredient name	4
P	Index position in A%	1
P1	Price	1
P2	Price of ingredient	1

P5	Total $/year for sale of project	3
PP	Total $/year for sale of ration	2
PT	Total $/year for sale of ingredient	1
Q	Quanity of ING sale	1
Q1	Quanity of SALE sale	2
QQ	Quanity of total project sale	3
RN	Ration number	2
TC	Counter for TOT array	1
TG	Pounds for TOT array	1
TT	TPROJ counter	2
U	Counter for printout array	3
U1	Unit number	1,2,3
X	Number of copies	
XB	Print format for Ingredient sales	1
XC	Print format for ration sales	2
XD	Print format for project sales	3
XX	Flag for month-to-date reports	1
Z	Loop indicator	4
ZZ	Flag to seperate closeouts	1,2

L. Description of individual program components

The three main routines in this program all operate in nearly the same manner. They all start with two main arrays and eventually end up with one array to print. The first routine will be explained in detail. Then the unique features of the others will be discussed.

Purpose: this routine sorts the month arrays and combines them with the years arrays to get the totals for the year and month. To start, you are asked how many copies are to be printed.

M. Solution procedure

Depending on the menu selection, the flags ZZ$ and XX are set. A statement on the screen alerts the operator to the importance of a backup at this point. The file, TOT is opened and the complete file is read into the TOT (KA, KC, KD, KE) arrays. TC counts the number of records in the file. At EOF, ING is opened and the complete file is read into the ING (K1, K3, K4, K5) arrays. Neither array set is sorted if there are no records present. The first sort is the ING arrays. CMD"0" is used to create an index file on the unit array (K1%). This groups all of the ingredient use by unit, one at the top, unit two next etc. This sort also sorts the array K3 but it sorts it by numeric value within the sorted K1 array. Thus 1-1 (unit 1, ingredient 1), 1-3, 1-4, are followed by 2-1, 2-4, 2-5. The index is set up on this sorted arrangement. This is an indirect sort because the physical order of the orginal arrays never change. Once an index is established on the ING arrays, an index array is developed for the TOT arrays, with the key sort on KA and the secondary sort on KC.

Reports are generated for each unit, listing month-to-date and year-to-date usage of each ingredient. The next array is designed so that totals are printed only for ingredients used. For example, if a unit only used corn in January and only soybean meal in Feburary, only corn and soybean meal would show up on the Febuary report. The Febuary report would show zero month-to-date corn usage, but the appropriate year-to-date corn usage.

After the indexes are set up, the ING arrays (K1 and K3) are are compared with KA and KC and merged together to form a third (AK - FK) array of a complete ordered list. The value of postion one of the indexed array A% is equal to P, and Q is the value of the indexed array B%. Five different comparisions are made between the two sets of arrays. P and Q keeps track of the positions in the arrays. First the unit numbers will be compared within the K1 and KA arrays. At position P and Q within the respective arrays, if the units are equal, than if the commodity numbers are equal (K3 and KC arrays), then access price, value and quantity data from the ING or TOT disk files, combine the two sets of arrays together and hold them in a third set of arrays (AK-GK). If the units are equal but the commodity numbers are not, then test for the smaller commmodity number and enter the result in the third array. If the files for either TOT or ING contains no records then the third array receives the total for the month or for the year only, depending on which one contains records. If the unit numbers are not equal and both have records in the files then the smaller unit number is found and saved in the third array. After the complete third array has been established, the reports are printed, starting a new page and printing a new header if the unit number changes. A fourth array is established to save the totals for the month if the operator has asked for month-end closeouts. Since in the third array there can be missing ingredients or some entries with just month information and others with just the years totals, before the printouts are run, another comparision must be made within the array, to predetermine the printout information for each entry. There are three formats; one with both month and year totals for each ingredient, one for just month totals and the third

for just year totals. The comparisions are made and this unit's information is printed followed by the unit's total pounds and total value for the month, and total pounds and total value for the year. The same printout procedure will follow in all the units.

If month-end closeout is requested the printout will be the same. After the complete ingredient printout, the information in the four arrays (X1-X5) is saved in the file TOT. The ingredient file is killed by CMD("KILL = ING").

The other two reports that are printed follow the same basic operation but sort on different record fields. The project reports sort on unit and project number. The ration reports sort on the unit number and ration number. The files involved in the project sort are SALE and TPROJ. The files for the ration sort are SALE and TRAT.

If month-end closeouts are requested, SALE and PURC are also killed with the months total being stored in TRAT and TPROJ.

This report generator takes around 10-15 minutes to go through the complete month-end reports.

IV. FUTURE UPDATES

Some suggestion for future updates could be to put the program on CP/M and use compiled BASIC. This could speed the calculations 10-fold.

The program could be put on a hard disk. This would expand the capacity of the program and would permit program addition to immediate screen access to data for management purposes.

A subroutine could be added to make ration price list.

A computer might be installed at the feedmill to allow immediate data entry. A terminal could be placed at the scales, and the rations formulations could be brought up on the screen for easy access by mill operators.

LITERATURE CITED

Apparat, Incorporated. 1981. NEWDOS/80 for the TRS-80 Model III Microcomputer. Denver, Colorado.

Standish, Thomas A. 1980. Data Structure Techniques. Addison-Wesley Publishing Company. pp. 133 - 139.

Tandy Corporation. 1980A. TRS-80 Model III Operation and BASIC Language Reference Manual. Fort Worth, Texas.

Tandy Corporation. 1980B. TRS-80 Model III Disk System Owner's manual. Fort Worth , Texas.

Tandy Corporation. 1980C. TRS-80 Model III Profile III Plus. Fort Worth, Texas.

Worland, Peter B. 1979. Introduction to BASIC Programming A Structured Approach. Houghton Mifflin Company. Dallas, Texas.

APPENDIX

APPENDIX TABLE A.1

INVENTORY LIST REPORT

INVENTORY

NUMBER	NAME	QUANITY	VALUE
1.	CORN	292602.00	$ 0.0626
2.	MILO	306552.00	$ 0.0535
3.	SBM	110972.00	$ 0.0984
4.	OATS	96757.60	$ 0.0609
5.	BRAN	1073.29	$ 0.0750
6.	WHEAT	0.00	$ 0.0000
7.	DEHY ALF	6143.47	$ 0.0498
8.	DICAL	4975.81	$ 0.1850
9.	LIMESTONE	2941.42	$ 0.0290
10.	STOCK SALT 167	4227.71	$ 0.0260
11.	Z-10 TM	312.40	$ 0.3600
12.	DRY MOLASSES	9.90	$ 0.1260
13.	TM SALT	2646.59	$ 0.0651
14.	SODIUM BICARB	2002.04	$ 0.2636
15.	AMM SO4	92.07	$ 0.1560
16.	KCL	542.74	$ 0.1380
17.	PULLET OYSTERS	511.30	$ 0.0680
18.	DIST SOLUBLES	21.08	$ 0.1100
19.	UREA	149.38	$ 0.1640
20.	BONEMEAL	270.54	$ 0.2800
21.	MONO NA PO4	244.91	$ 0.4250
22.	MAG O2	567.23	$ 0.2300
23.	COTTON SEED HUL	662.13	$ 0.0792
24.	ANIMAL FAT	1.53	$ 0.2100

APPENDIX TABLE A.2

COMPLETE RATION LISTING BY UNIT

UNIT LISTING

1 . RATION ID - HORSE OATS

NO.	ID#	INGREDIENT	PERCENT
1	4	OATS	100.00
			100

2 . RATION ID - 82-14% HORSE FEED

NO.	ID#	INGREDIENT	PERCENT
1	1	CORN	58.90
2	4	OATS	16.90
3	3	SBM	11.10
4	9	LIMESTONE	1.40
5	13	TM SALT	0.10
6	25	KSU SWINE PRMX.	0.10
7	5	BRAN	7.50
8	30	WET MOLASSES	4.00
			100

APPENDIX TABLE A.3

MONTH-TO-DATE PURCHASES

PURCHASES

	DATE	NO.	INGREDIENT	QUANITY	VALUE($/LB)	TOTAL
1	07/01/83	14	SODIUM BICARB	600.000	0.2600	156.0000
2	07/01/83	21	MONO NA PO4	100.000	0.4250	42.5000
3	07/01/83	71	PURE PRIDE 200	500.000	0.1050	52.5000
4	07/02/83	9	LIMESTONE	500.000	0.0290	14.5000
5	07/02/83	14	SODIUM BICARB	500.000	0.2600	130.0000
6	07/02/83	13	TM SALT	250.000	0.0650	16.2500
7	07/02/83	16	KCL	150.000	0.1350	20.2500
8	07/02/83	8	DICAL	100.000	0.1850	18.5000
9	07/05/83	8	DICAL	400.000	0.1850	74.0000
10	07/05/83	22	MAG O2	100.000	0.2300	23.0000
11	07/07/83	8	DICAL	1000.000	0.1850	185.0000
12	07/08/83	8	DICAL	500.000	0.1850	92.5000
13	07/08/83	10	STOCK SALT 167	1500.000	0.0260	39.0000
14	07/08/83	9	LIMESTONE	500.000	0.0290	14.5000
15	07/08/83	10	STOCK SALT 167	500.000	0.0260	13.0000
16	07/08/83	5	BRAN	500.000	0.0750	37.5000
17	07/09/83	8	DICAL	500.000	0.1850	92.5000
18	07/09/83	17	PULLET OYSTERS	200.000	0.0680	13.6000
19	07/09/83	13	TM SALT	150.000	0.0650	9.7500
20	07/09/83	20	BONEMEAL	100.000	0.2800	28.0000
21	07/09/83	23	COTTON SEED HUL	700.000	0.0810	56.7000
22	07/11/83	13	TM SALT	150.000	0.0650	9.7500
23	07/11/83	9	LIMESTONE	750.000	0.0290	21.7500
24	07/11/83	10	STOCK SALT 167	500.000	0.0260	13.0000
25	07/11/83	8	DICAL	500.000	0.1850	92.5000
26	07/11/83	16	KCL	300.000	0.1350	40.5000

APPENDIX TABLE A.4

MONTH-TO-DATE SALES

SALES UP TO DATE FOR THE MONTH

DATE	UNIT#	TYPE	ID#	QUANITY	PROJECT#	KIND	$/LB	TOT/COST
06/30/83	1	R	2	12435.000		B	0.0743	923.92
06/30/83	1	R	3	500.625		S	0.0842	42.15
06/30/83	2	I	13	10.000		B	0.0729	0.73
06/30/83	2	I	21	10.000		B	0.4760	4.76
06/30/83	2	R	8	150.000		S	0.1711	25.67
06/30/83	2	R	6	1842.300		B	0.0776	142.96
06/30/83	3	R	4	24000.000		S	0.0796	1910.40
06/30/83	3	R	4	19000.000		B	0.0756	1436.40
06/30/83	3	R	2	12000.000		B	0.0710	852.00
06/30/83	3	R	44	9975.000		S	0.1098	1095.26
06/30/83	3	R	56	4000.000		S	0.0876	350.40
06/30/83	3	R	35	7000.000		S	0.0804	562.80
06/30/83	4	R	9	20700.000		B	0.0719	1488.33
06/30/83	4	I	23	50.000		B	0.0887	4.44
06/30/83	4	R	23	8108.000		B	0.0733	594.32
06/30/83	5	R	22	1035.000		B	0.0721	74.62
06/30/83	5	R	16	5025.000		B	0.1109	557.27
06/30/83	5	R	28	1000.000		S	0.0657	65.70
06/30/83	5	R	29	1000.000		S	0.0657	65.70
06/30/83	5	R	30	1000.000		S	0.0657	65.70
06/30/83	5	R	31	1000.000		S	0.0657	65.70
06/30/83	5	R	32	1000.000		S	0.0657	65.70
06/30/83	5	R	33	1000.000		S	0.0657	65.70
06/30/83	5	R	34	1000.000		S	0.0657	65.70
06/30/83	5	R	35	1000.000		S	0.0657	65.70

APPENDIX TABLE A.5

RATION ID AND NUMBER BY UNIT

UNIT LISTING

****

1 . HORSE OATS
2 . 82-14% HORSE FEED
3 . 82-18% FOAL CREEP
4 . 83-20% FOAL CREEP

APPENDIX TABLE A.6

MONTH-TO-DATE AND MONTH-END CLOSEOUTS INGREDIENT USE REPORT

INDIVIDUAL INGREDIENT LIST

UNIT #1 - HORSE UNIT

NUM	NAME	LBS/MO	$/LB	$/MO	LB/YR	$/YR
1	CORN	7953.48	0.0704	559.867	7953.48	559.8670
3	SBM	1804.95	0.1116	201.411	1804.95	201.4110
4	OATS	2436.18	0.0687	167.352	2436.18	167.3520
8	DICAL	21.25	0.2183	4.639	21.25	4.6392
9	LIMESTONE	200.24	0.0325	6.512	200.24	6.5123
13	TM SALT	24.94	0.0749	1.867	24.94	1.8668
20	BONEMEAL	895.32	0.0840	75.207	895.32	75.2069
25	KSU SWINE PRMX.	16.79	0.3861	6.481	16.79	6.4814
30	WET MOLASSES	564.87	0.0570	32.181	564.87	32.1809
61	MINERAL BLKS	500.00	0.0761	38.050	500.00	38.0500

TOTALS

LBS/MO - 14418
$/MO - 1093.57
LBS/YR - 14418
$/YR - 1093.57

APPENDIX TABLE A.7

MONTH-TO-DATE AND MONTH-END CLOSEOUTS PROJECT NUMBER USE REPORT

UNIT #1 - HORSE UNIT

PROJECT#	LB/MO	$/LB	$/MO	LB/YR	$/YR
	14443.100	0.0757	1093.590	14443.10	1093.5900

******

LBS/MO - 14443.1
$/MO - 1093.59
LBS/YR - 14443.1
$/YR - 1093.59

APPENDIX TABLE A.8

MONTH-TO-DATE AND MONTH-END CLOSEOUTS RATION USE REPORT

UNIT #1 - HORSE UNIT

RATION #	LB/MO	$/LB	$/MO	LB/YR	$/YR
2	12435.00	0.0743	923.921	12435.00	923.9210
3	500.63	0.0842	42.153	500.63	42.1526
4	1007.50	0.0888	89.466	1007.50	89.4660

TOTALS

LBS/MO - 13943.1
$/MO - 1055.54
LBS/YR - 1007.5
$/YR - 1055.54

APPENDIX TABLE A.9

ALPABETIZED INVENTORY LIST

COMMODITY #	NAME
59	AH-119
15	AMM SO4
48	AMPROL 25
78	AN PREMIX
24	ANIMAL FAT
65	ARSANILIC ACID
98	ASP-250
43	AUREO 50
42	AUREO10
93	BACIFERM 10
67	BAN-A-WORM
68	BANMITH
36	BLOAT GAURD BLK
33	BLOCK SALT
79	BN PREMIX
55	BOLSEN DRY CORN
56	BOLSEN WET CORN
20	BONEMEAL
62	BOVATEC

APPENDIX B.1

MAIN PROGRAM LISTING

```
REM This program was written by Coree L. Smith, 1983-1984.  It is a program
      written for feed manufacturing personal at a university to help in the
      billing of research units sales and purchases.  The REM outside the BASIC
      program will not be recognized by the BASIC programming language.
1 CLS:CLEAR 4500
5 POKE 16425,0:POKE 16424,67
6  CLOSE
7 OPEN "R",1,"INV","FF",36:IF EOF(1) THEN GOTO 8 ELSE X9=LOF(1)
8 DIM IN$(20),IN%(20),PC$(20),PC(20),WT(20),WT$(20)
9 DIM DF$(20),DE$(20),IV$(100),DA(20),DB$(20),DC$(20),DD$(20):GOSUB 50000
10 POKE 16916,0:CLS:
17 CLOSE
20 PRINT CHR$(23)
30 POKE 16916,20
40 PRINT @10,"KANSAS STATE UNIVERSITY
50 PRINT @84,"MAIN MENU
55 YY=0
60 PRINT
70 CLS
REM *** Main Menu ***********************************************
80 PRINT @202 , "1. ADD INGREDIENTS
90 PRINT @266,"2. EDIT,DELETE INGREDIENTS
100 PRINT @ 330,"3. ADD RATIONS
110 PRINT@394,"4. EDIT,DELETE RATIONS
120 PRINT @458,"5. REPORTS
130 PRINT @522,"6. DAILY SALES
140 PRINT@586,"7. DAILY PURCHASES
143 PRINT @650,"8. PURC & SALE CORRECTIONS
145 PRINT @714,"9. EXIT PROGRAM
160 PRINT @ 842,"ENTER NUMBER";
170 INPUT B2$
171 Z$=B2$:GOSUB 40000
172 IF Z=1 THEN GOTO 70
180 POKE16916,0
185 CLS
186 AA$="0"
188 IF B2$="H" OR B2$="HELP" THEN GOSUB 6000
189 IF B2$="3" THEN A$="1":YY=1
190 IF B2$="4" THEN A$="2":AA$="2":W=0:WW=0
192 B2=VAL(B2$)
196 IF B2=1 THEN GOSUB 500
197 IF B2=2 THEN GOSUB 780
198 IF B2=3 THEN GOSUB 3000
199 IF B2=4 THEN GOSUB 3000
201 IF B2=5 THEN GOSUB 10000
202 IF B2=6 THEN GOSUB 4000
203 IF B2=7 THEN GOSUB 7000
204 IF B2=8 THEN GOSUB 62000
```

```
205 IF B2=9 THEN GOSUB 220
210 GOTO 10
REM *** Backup Procedure**************************************
220 CLS:POKE 16916,3
230 PRINT @ 30,"BACKING UP
240 PRINT @64,STRING$(64,"*")
250 PRINT @138,"ENTER BACKUP DISK 0 IN TOP DRIVE(DRIVE #1)."
260 PRINT @202,"HIT ENTER WHEN COMPLETED OR <M> FOR
     MAIN/MENU";:INPUT A$
265 IF A$="M" THEN RETURN
266 IF A$<>"" THEN GOTO 220
267 CLEAR 100
270 CLS:CMD"COPY 0 1 09/09/83"
280 POKE 16916,0:CLS:POKE 16916,4
290 PRINT @ 25 ,"BACKING UP DISK #1"
300 PRINT@64,STRING$(64,"*")
310 PRINT @138,"ENTER  BACKUP DISK #1 IN TOP DRIVE(DRIVE #1)"
320 PRINT @202,"HIT ENTER TO CONTINUE(FOLLOW DIRECTION THAT
     FOLLOW)";:INPUT A$
330 CMD"COPY 0 1 09/09/83"
340 POKE 16916,0:CLS
350 PRINT CHR$(23):PRINT @210,"THE END"
355 PRINT @384,"REMOVE DISK AND TURN MACHINE OFF"
360 END
REM *** Add ingredients subroutine ****************************
500 CLS
510 OPEN "R",1,"INV","FF",36
520 PRINT:PRINT:PRINT
530 POKE 16916,3:PRINT@23, "ADD INGREDIENTS"
540 PRINT @74,"ENTER <E>DIT,<M>AIN,<ENTER> TO CONTINUE
550 A1$=STRING$(64,"*"):PRINT@128,A1$
560 X9=X9+1:PRINT@340,"ENTER COMMODITY #: ";X9
565 PRINT@340,"ENTER COMMODITY #: ";X9
570 CN$=STR$(X9)
580 PRINT@404,"ENTER COMMMODITY NAME: ";
590 LINE INPUT N$
595 IF N$="M" THEN X9=X9-1:GOTO 760
600 PRINT@468,"ENTER POUNDS: ";
610 LINE INPUT LB$
615 Z$=LB$:GOSUB 40000:IF Z=1 THEN PRINT CHR$(27);CHR$(31):GOTO 600
620 PRINT@532,"ENTER INITIAL VALUE($/LB$): ";
630 LINE INPUT P$
631 Y1=0
634 Y1$=P$:GOSUB 40000:IF Y1=1 THEN 620
635 IF VAL(P$)>1.00 THEN PRINT "IS THIS PRICE CORRECT ON A DOLLARS
     PER LB BASIS,(Y OR N)":INPUT M$:Y1$=M$:GOSUB 40300:IF Y1=2 THEN
     PRINT @530,"";CHR$(31):GOTO 620
REM *** New screen prints ingredients information *************
640 CLS
650 PRINT@197,"NAME        NUMBER     QUANITY     VALUE"
660 PRINT@261,N$
670 PRINT @282,CN$
676 CN=VAL(CN$)
```

```
680 PRINT@296,LB$
690 PRINT @312,P$
700 PUT 1,CN,1140
702 Z9$="":PRINT @350,"";:INPUT Z9$:CLS
710 IF Z9$="" THEN GOTO 560
720 IF Z9$="E"THEN GOSUB 820:CLS
730 IF Z9$="M"THEN GOTO760
735 PRINT @714,"ONLY  ENTER <E>,<M>,OR <ENTER>,REDO":GOTO 702
740 R=0
750 CLOSE
760 POKE 16916,0
770 RETURN
780 REM INVEN FILE EDITOR ************************************
790 OPEN "R",1,"INV","FF",36
800 PRINT@260,"ENTER COMMODITY <NUMBER> TO BE EDITED ";
810 LINE INPUT CN$
811 IF CN$ = "M" THEN RETURN
815 Z$=CN$:GOSUB 40000:IF Z=1 THEN 800
820 CN = VAL(CN$)
830 IF CN>X9 THEN CLS:PRINT @ 74,"INGREDIENT NUMBER ";CN;" IS TO
      LARGE(<ENTER>)";:INPUT A$:RETURN
850 GET 1,CN,1140
860 CLS
870 PRINT"COMMODITY NUMBER = ";CN
880 PRINT"COMMODITY NAME = ";N$
890 PRINT"QUANITY = ";LB$
900 PRINT"VALUE ($/LB)= ";P$
910 PRINT
920 PRINT"1. CHANGE ALL"
930 PRINT"2. CHANGE PRICES ONLY
940 PRINT"3. CHANGE POUNDS ONLY
942 PRINT "4. CHANGE NAME ONLY
950 PRINT"5. DELETE ITEM FORM FILE
960 PRINT"6. NO CHANGE FOR THIS DATA
980 LINE INPUT "ENTER YOUR CHOICE FROM THE OPTIONS ABOVE : ";R$
990 IF VAL(R$)<1 OR VAL(R$)>6 THEN  PRINT "ENTER NUMBER 1-6 ONLY
      PLEASE ": GOTO 860
1000 IF VAL(R$) =1 THEN GOSUB 1100: GOSUB 1120:GOSUB 1131
1010 IF VAL(R$) = 2 THEN GOSUB 1120
1020 IF VAL(R$) = 3 THEN GOSUB 1100
1025 IF VAL(R$)=4 THEN GOSUB 1131
1030 IF VAL(R$) = 5 THEN CLS:PRINT "ARE YOU SURE YOU WANT TO
      DELETE THIS FILE":INPUT M$:Y1$=M$:GOSUB 40300:IF Y1=1 THEN
      CN$="":LB$="":N$="":P$=""
1040 IF VAL(R$) = 6 THEN  RETURN
1045 PRINT
1050 PUT 1,CN,1140:GOTO 860
1070 PRINT"END OF DATA IN THIS FILE
1080 CLOSE
1090 RETURN
1100 LINE INPUT "ENTER CORRECTED # OF LBS : ";LB$
1110 RETURN
1120 LINE INPUT "ENTER CORRECTED PRICE : ";P$
```

```
1130 RETURN
1131 LINE INPUT "ENTER CORRECTED NAME: ";N$:RETURN
1135 REM FILE FORMATS**********************************
1140 (5)CN$,(15)N$,(10)LB$,(6)P$;
1145 REM TOTAL SALES / MONTH
1150 (10)D1$,(2)U1$,(1)G$,(5)B1$,(10)Q1$,(15)PJ$,(2)S1$,(6)P1$;
1160 (8)D2$,(5)I1$,(10)Q2$,(6)V1$,(10)P3$;
1165 REM IND INGREDIENT SALES
1170 (5)U1$,(6)CN$,(12)Q$,(15)N$,(6)P2$;
1180 (5)LS$,(8)MO$,(10)TT$;
1185 (3)X9,(3)H1,(3)H2,(3)H3,(3)H4,(3)H5,(3)H6,(3)H7,(3)H8,(3)H9,(3)H0,(10)D1$;
1186 (10)DE$,(10)DF$;
2000 Z1$=""
2010 PRINT
2020 REM *** ROUTINE TO ADD, DELETE AND EDIT RATIONS***********
2040 CLS:A$="##  #####   %       % ###.####"
2060 POKE 16916,3
2070 TL=0:GOSUB 2320
REM *** Entering the ration **********************************
2080 PRINT@896,"";:INPUT"RATION ID ";RI$:GOSUB 2330
2090 PRINT@896,"INGRED. NO. ";CHR$(30);:INPUT IN$:IF VAL(IN$)>X9THEN
     GOTO 2090ELSE Z$=IN$:GOSUB 40000:IF Z=1 THEN 2090
2091 IF IN$="@" THEN N2%=N%-1:GOTO 2120:ELSE IN%(N%)=VAL(IN$)
2092 IF IN$="M" THEN O$="M":GOTO 2180
2095 IF IN$<>"@" THEN PRINT@926,"POUNDS";:INPUT
     WT$(N%):Z$=WT$(N%):GOSUB 40000:IF Z=1 THEN PRINT@896,"INGRED. NO.
     ";IN$;CHR$(30):GOTO 2095
2096 WT(N%)=VAL(WT$(N%))
2100 IF N1%=<10 THEN GOSUB 2310:ELSE N1%=1:CLS:GOTO 2100
2115 TL=TL+WT(N%):PRINT@42,CHR$(30);TL;:N1%=N1%+1:N%=N%+1:GOTO 2090
2116 IF AA$="2" THEN TP=0:N1%=1:FOR N%=1TO N2%:TP=TP+PC(N%):GOTO
     2130
2120  IF AA$="2" THEN GOTO 2116:ELSETP=0:N1%=1:FOR
     N%=1TON2%:PC(N%)=(WT(N%)/TL)*100:TP=TP+PC(N%)
2130 E$="":IF N1%=<10 THEN GOSUB 2310:ELSE
     PRINT@896,CHR$(30);N2%-(N1%-1);" INGREDIENTS REMAIN. <E>NTER TO
     CONTINUE.";:INPUT E$:IF E$=""THEN N1%=1:CLS:GOTO 2130:ELSE 2130
2140 N1%=N1%+1:NEXT N%:PRINT@42,CHR$(30);TL;:PRINT @52,"";:PRINT USING
     "###.###";TP;:PRINT "%";
2145 IF IN$="M"  ORTP<99 OR TP>101 THEN PRINT@896,CHR$(30);"EDIT YOUR
     PERCENT DOESN'T EQUAL 100%";:INPUT A$:O$="E":GOTO 2190
2150 O$="":PRINT@896,CHR$(30);"<C>ONTINUE,<S>AVE, <R>EVIEW, <E>DIT,
     RETURN TO <M>ENU";:INPUT O$
2160 IF O$="S" THEN 2200
2170 IF O$="R" THEN 2270
2180 IF O$="M" THEN CLOSE:GOTO 2300
2185 IF O$="C" THEN IF AA$="2"THEN GOTO 2150:ELSEPOKE
     16916,0:CLS:GOTO2400
2190 IF O$="E" THEN M%=1: GOTO  2280:ELSE 2150
REM *** Saving the Ration ************************************
2200  Z1$="":CLS:FOR K%=1TO N2%
2210   IN$(K%)=STR$(IN%(K%))
2220   PC$(K%)=STR$(PC(K%))
```

```
2230   Z1$=Z1$+IN$(K%) + PC$(K%)
2240   NEXT K%
2245 Z1$=RI$+"Z"+Z1$
2250 IF YY<>1   AND M%=1 THEN PUT 1,$,,Z1$;:GOTO 2270
2260 PUT 1,,,Z1$;
2270 GOSUB 2320:GOTO 2120
REM *** Edit the ration ***************************************
2280 PRINT@896,CHR$(30);:INPUT "EDIT NO. ";N$:Z$=N$:GOSUB
     40000:N%=VAL(N$):IF Z=1 THEN 2280
2281 IF N$="M" THEN RETURN
2282 IF N%>N2% THEN 2280:ELSE PRINT@910,"";:INPUT "INGRED. NO.
     ";IN$(N%):IF VAL(IN$(N%))>X9 THEN PRINT @910,"",:INPUT "INGRED. NO.
     ";IN$(N%):ELSE Z$=IN$(N%):GOSUB 40000:IN%(N%)=VAL(IN$(N%)):IF Z=1
     THEN GOTO 2282
2283  IF AA$="2" THEN PRINT @930,"";: INPUT "PERCENT
     ";PC$(N%):Z$=PC$(N%):GOSUB 40000:PC(N%)=VAL(PC$(N%)):IF Z=1 THEN
     2282
2284 IF AA$<>"2" THEN PRINT @930,"";:INPUT "POUNDS
     ";WT$(N%):Z$=WT$(N%):GOSUB 40000:WT(N%)=VAL(WT$(N%)):IF Z=1 THEN
     2282
2285 IF AA$<>"2" THEN FOR N%=1TON2%:TL=TL+WT(N%):NEXT N%
2290 TP=0:TL=0:FOR N%=1 TO N2%:TL=TL+WT(N%):NEXT N%:GOTO 2270
2300 POKE 16916,0:RETURN
2310 NA$=IV$(IN%(N%)):A$="##  #### %       % #####.###
     ###.####":PRINT@128+(N1%*64),"";:PRINT USING
     A$;N%,IN%(N%),NA$,WT(N%),PC(N%):RETURN
REM *** Review the ration *************************************
2320 :PRINT@0,CHR$(31);"RATION ADD, DELETE, EDIT    TOTAL LBS "
2330 :PRINT@75,STRING$(6," ");@64,"RATION ID ";@74,CHR$(30);RI$:N%=1:N1%=1
REM *** Continuation subroutine **********************************
2340 PRINT @128,"NO.   ID#  INGREDIENT     LBS   PERCENT":RETURN
2400 CLS:PRINT @30,"MORE RATIONS":PRINT @74,"1. SAME UNIT":PRINT
     @138,"2. DIFFERENT UNIT":PRINT @202,"3. MAIN MENU
2405 PRINT @266,"ENTER NUMBER";:INPUT ZX$
2410 IF VAL(ZX$)=1 THEN W=1:WW=1:GOTO 2000
2420 IF VAL(ZX$)=2 THEN A$="1":GOTO 3000
2425 IF ZX$="3" OR ZX$="M"ORZX$="MAIN" THEN O$="M" :GOTO 2180
2430 GOTO 2400
3000 CLS:
3005 Q%=0
REM *** Prints the unit Names ************************************
3010 PRINT @89,"UNITS
3020 PRINT @128,STRING$(64,"*")
3030 PRINT@195,"1. HORSE UNIT";:PRINT@225,"9. DAIRY UNIT
3040 PRINT @259,"2. SHEEP UNIT ";:PRINT @289,"10. POULTRY UNIT
3050 PRINT@ 323,"3. SWINE UNIT";:PRINT@353,"11. CASSODAY COWS
3060 PRINT@ 387,"4. PUREBRED UNIT";:PRINT@417,"12. KONZA COWS
3070 PRINT@ 451,"5. BEEF RESEARCH UNIT";:PRINT@481,"13. BARTLEY'S
     COWS
3080 PRINT@ 515,"6. GRASS UNIT";:PRINT@545,"14. SPARE 1
3090 PRINT@ 579,"7. SHORTHORN UNIT";:PRINT@609,"15. SPARE 2
3100 PRINT@643,"8. COW-CALF UNIT"
3130 PRINT @896,STRING$(64,"*")
```

```
3140 INPUT "ENTER UNIT NUMBER "; M$:Z$=M$:GOSUB 40000:IF Z=1 THEN
     3000
3145 IF M$="M" OR M$="MAIN" THEN RETURN
3150 M=VAL(M$)
3160 IF M<1 OR M>15 THEN PRINT "TRY AGAIN ONLY NUMBERS BETWEEN 1
     AND 15, ENTER TO REDO":INPUT L$:GOTO 3000
3168 A=VAL(A$)
3169 CLOSE
REM *** Opens ration files ************************************
3170 IF A=1 THEN ON M GOTO
     3330,3310,3290,3320,3280,3300,3270,3350,3355,3340,3351,3352,3353,3356,335
     7:ELS E ON M GOTO
     3460,3440,3420,3450,3410,3430,3400,3480,3490,3470,3491,3492,3493,3494,349
     5
3270 OPEN "E",1,"MU/SH","MU",300:GOTO 3360
3280 OPEN "E",1,"MU/BF","MU",300:GOTO 3360
3290 OPEN "E",1,"MU/SW","MU",300:GOTO 3360
3300 OPEN "E",1,"MU/GR","MU",300:GOTO 3360
3310 OPEN "E",1,"MU/SP","MU",300:GOTO 3360
3320 OPEN "E",1,"MU/PB","MU",300:GOTO 3360
3330 OPEN "E",1,"MU/H","MU",300:GOTO 3360
3340 OPEN "E",1,"MU/PT","MU",300:GOTO 3360
3350 OPEN "E",1,"MU/COW","MU",300:GOTO 3360
3351 OPEN "E",1,"MU/CA","MU",200:GOTO  2000
3352 OPEN "E",1,"MU/KON","MU",200:GOTO 2000
3353 OPEN "E",1,"MU/BT","MU",200:GOTO 2000
3355 OPEN "E",1,"MU/D","MU",300:GOTO 3360
3356 OPEN "E",1,"MU/SPA","MU",200:GOTO 2000
3357 OPEN "E",1,"MU/SPB","MU",200:GOTO 2000
3360 GOTO 2000
3400 OPEN "R",1,"MU/SH","MU",300:GOTO 3500
3410 OPEN "R",1,"MU/BF","MU",300:GOTO 3500
3420 OPEN "R",1,"MU/SW","MU",300:GOTO 3500
3430 OPEN "R",1,"MU/GR","MU",300:GOTO 3500
3440 OPEN "R",1,"MU/SP","MU",300:GOTO 3500
3450 OPEN "R",1,"MU/PB","MU",300:GOTO 3500
3460 OPEN "R",1,"MU/H","MU",300:GOTO 3500
3470 OPEN "R",1,"MU/PT","MU",300:GOTO 3500
3480 OPEN "R",1,"MU/COW","MU",300:GOTO 3500
3490 OPEN "R",1,"MU/D","MU",300:GOTO 3500
3491 OPEN "R",1,"MU/CA","MU",200:GOTO 3500
3492 OPEN "R",1,"MU/KON","MU",200:GOTO 3500
3493 OPEN "R",1,"MU/BT","MU",200:GOTO 3500
3494 OPEN "R",1,"MU/SPA","MU",200:GOTO 3500
3495 OPEN "R",1,"MU/SPB","MU",200:GOTO 3500
REM *** Beginning ration EDIT *********************************
3500 IF W=0 AND WW=0 THEN CLS:PRINT @74," ":INPUT "WHAT RATION
     NUMBER DO YOU WANT TO SEE?";B$:GOSUB 3800:Z$=B$:GOSUB 40000:IF
     Z=1 THEN GOTO 3500
3502 IF W<>0 AND WW<>0 THEN B$=B1$
3503 IF WW=1 THEN GOTO 3515
3505 B=VAL(B$)
REM *** Loading the ration ************************************
```

```
3509 FOR I=1 TO B
3511 GET 1,,,Z1$;:PRINT @60,"*"
3512 NEXT I:GOTO 3525
3515 PRINT
3517 IF EOF(1) THEN RETURN ELSE I=I+1:GET 1,,,Z1$;
3525 CA$="":CO=1
3526 CC$=MID$(Z1$,CO,2):CB$=MID$(Z1$,CO,1):IF CB$="Z" AND CC$="Z " THEN
     GOTO 3530
3527 CA$=CA$+CB$:CO=CO+1:GOTO 3526
3530 QZ=LEN(CA$)+2
3531 Q%=QZ-1
3532 C$=MID$(Z1$,Q%)
3533 CJ=LEN(C$)
3535 N2%=0
3537 T%=2
3550 N2%=N2%+1
3551 H$=""
3553 T%=T%+1
3555 D$=MID$(C$,T%,1)
3560 IF D$=" " THEN  GOTO 3580
3565 H$=H$+D$
3566 PRINT @60,"*"
3570 GOTO 3553
3580 IN$(N2%)=H$
3583 H$=""
3585 T%=T%+1
3610 D$=MID$(C$,T%,1)
3620 IF T%>CJ OR D$=" " THEN GOTO 3650
3630 H$=H$+D$
3640 GOTO 3585
3650 PC$(N2%)=H$
3660 IF T%+1<CJ THEN 3550
3670 CLS:FOR K%=1 TO N2%
3680    IN%(K%)=VAL(IN$(K%))
3690    PC(K%)=VAL(PC$(K%))
3695 PRINT @60,"*"
3700    NEXT K%
3703 RI$=CA$
3705 IF W=1 OR WW=1 THEN RETURN
3710 GOTO 2270
4000 REM *** DAILY RATION AND INGREDIENT SALES ********************
4005 CLS
4006 PO=1
4010 PRINT @30,"SALES"
4020 PRINT @64,STRING$(64,"*")
4025 POKE 16916,2
4026 PRINT @202,"1. DATE (MO/DA/YR) - ";D1$;: PRINT @ 221,"";
4027 T1$=INKEY$:IF T1$="" THEN 4027
4028 IF T1$="Y" THEN GOTO 4050
4030 PRINT @202,"";
4040 LINE INPUT"1. DATE (M0/DA/YR) - ";D1$
4045 Z$=D1$:GOSUB 45000:IF Z=1 THEN PRINT@200,CHR$(31):GOTO 4030
4050 PRINT @266,"";
```

```
4060  LINE INPUT "2. UNIT# - ";U1$
4065 Z$=U1$:GOSUB 40000:IF Z=1 THEN PRINT @ 260,CHR$(31):GOTO 4050
4066 IF VAL(U1$)>15 OR VAL(U1$)<1 THEN PRINT@260,CHR$(31):GOTO 4050
4070 PRINT@330,"";
4080 LINE INPUT "3.TYPE(<R>ATION OR <I>NGREDIENT) - ";G$
4085 Z$=G$:IF G$<>"R" AND G$<>"I" THEN PRINT@984,CHR$(31):GOTO 4085
4090 PRINT@394,"";
4100 LINE INPUT "4. RATION OR INGREDIENT # - ";B1$
4101 Z$=B1$:GOSUB 40000:IF Z=1 THEN PRINT @390,CHR$(31):GOTO 4090
4102 IF G$="R" THEN B$=B1$:M=VAL(U1$):GOSUB 3800:IF VV=1 THEN GOTO
     4000:ELSE IF VV=2 THEN RETURN:
4110 PRINT @458,"";
4120 LINE INPUT "5. QUANITY - ";Q1$
4121 Z$=Q1$:GOSUB 40000:IF Z=1 THEN PRINT@456,CHR$(31):GOTO 4110
4130 PRINT @522,"";
4140 LINE INPUT "6. PROJECT #  (OPTIONAL) - ";PJ$
4150 PRINT @586,"";
4160 LINE INPUT "7. <B>ULK OR <S>ACK - "; S1$
4161 IF MID$(S1$,1,1)<>"S" AND MID$(S1$,1,1)<>"B" THEN
     PRINT@584,CHR$(31):GOTO 4150
4170 PRINT @778,"";
4180 PRINT "ARE THESE CORRECT?(<Y>ES OR <N>O) - "
4185 T1$=INKEY$:IF T1$="" THEN 4185
4186 T1=VAL(T1$)
4187 IF T1$="Y" THEN 4191
4188 IF T1$="N" THEN 4201
4189 PRINT "ONLY ENTER Y OR N TRY AGAIN":INPUT J$:PRINT
     @776,CHR$(31):GOTO 4170
4190 IF J$="N" OR J$="NO" THEN GOTO 4210
4191 M$=U1$:
4192 IF G$="R" THEN GOSUB 5500
4194 IF G$="I" THEN GOSUB 5000
4200 GOTO 4219
4201 REM VALUE BEING CHANGED*****************************
4202 PRINT @770,"WHICH VALUE NEEDS CORRECTED(ENTER NUMBER) ";
4203 INPUT J1$:Z$=J1$:GOSUB 40000:IF Z=1 THEN PRINT @769,CHR$(31):GOTO
     4202
4204 IF VAL(J1$)<1 AND VAL(J1$)>7 THEN PRINT @768,CHR$(31):GOTO 4202
4205 IF VAL(J1$)=1 THEN LINE INPUT "CORRECTED DATE - ";D1$
4206 J1=VAL(J1$):IF J1<1 OR J1>7 THEN PRINT @769,CHR$(31):GOTO 4201
4207 IF J1=2 THEN LINE INPUT "CORRECT UNIT NUMBER - ";U1$
4208 IF J1=3 THEN LINE INPUT "CORRECT TYPE <R> OR <I> - ";G$
4209 IF J1=4 THEN LINE INPUT " CORRECT ID NUMBER - ";B1$
4210 IF J1=5 THEN LINE INPUT " CORRECT QUANITY - ";Q1$
4211 IF J1=6 THEN LINE INPUT " CORRECT PROJECT # - ";PJ$
4212 IF J1=7 THEN LINE INPUT " CORRECT <B>ULK OR <S>ACK - ";S1$
4213 CLS:PRINT@202,"1. DATE -  ";D1$
4214 PRINT @ 266,"2. UNIT # - ";U1$:PRINT @ 330,"3. TYPE - ";G$
4215 PRINT @ 394,"4. ID # - ";B1$
4216 PRINT @458,"5. QUANITY - ";Q1$:PRINT @522,"6. PROJECT # - ";PJ$
4217 PRINT @586,"7. <B>ULK OR <S>ACK - ";S1$
4218 GOTO 4170
4219 U1=VAL(U1$):IF PO = 0 THEN GOTO 4260
```

```
REM *** Storing the sale ***************************************
4220 CLOSE : OPEN "E",2,"SALE:1","FF",51
4225 PRINT P1:P1$=STR$(P1)
4230 PUT 2,,1150
4250 CLOSE
4260 CLS:PRINT@960,"";
4270 LINE INPUT "PRESS ENTER TO CONTINUE OR <M>AIN MENU - ";J2$
4280 IF J2$="" THEN GOTO 4005:ELSE RETURN
5000 REM SALES ROUTINE INGREDIENTS
5010 CLOSE 1:OPEN "R",1,"INV","FF",36
5015 CLOSE 2:OPEN "E",2,"ING:1","FF",44
5020 B1=VAL(B1$)
5030 GET 1,B1,1140
5035 IF VAL(LB$)<VAL(Q1$) THEN CLS:PRINT @ 134,"THIS SALE CANN'T BE
     COMPLETED THERE IS NOT ENOUGH ";N$;:INPUT A$:PO=0:GOTO 5060
5040 P1=VAL(Q1$)*VAL(P$)
5045 IF S1$="S" THEN P1=P1*1.18/VAL(Q1$)
5046 IF S1$="B" THEN P1=P1*1.12/VAL(Q1$)
5050 LB=VAL(LB$)-VAL(Q1$)
5053 Q = VAL(Q1$)
5054 Q$=STR$(Q):CN$=B1$
5055 LB$=STR$(LB):P1$=STR$(P1)
5056 P2$=STR$(P1):PUT 1,$,1140:GOSUB 8400
5060 RETURN
5500 REM *** RATION SALE SUB ***********************************
5503 FG=0
5505 CLS:PRINT@128,"PLEASE BE PATIENT"
5506 A=0:W=1:M$=U1$:WW=2:GOSUB 3150
5520 P1=0:P1$="":P2=0
5525 CLOSE : OPEN "R",1,"INV","FF",36
5526 OPEN "E",2,"ING:1","FF",44
5530 FOR I%=1 TO N2%
5540 CN=VAL(IN$(I%))
5550 GET 1,CN,1140
5555 PRINT P$
REM *** FG = 1 If sale is being corrected ***********************
5560 Q=VAL(Q1$)*(VAL(PC$(I%))*.01):IF FG=1 THEN Q=0-Q
5561 IF Q>VAL(LB$) THEN CLS:PRINT @130,"THERE IS NOT ENOUGH ";N$;" TO
     COMPLETE THIS RATION";:INPUT A$:PO=0
5580 LB=VAL(LB$)-Q
5582 IF S1$="S" OR S1$="S "THEN P2=Q*VAL(P$)*1.18
5584 IF S1$="B"  OR S1$="B " THEN P2=Q*VAL(P$)*1.12
5590 P1=P2+P1
5591 P2=P2/Q
5594 Q$=STR$(Q):CN$=STR$(CN)
5595 LB$=STR$(LB):P1$=STR$(P1)
5596 DA(I%)=CN:DB$(I%)=N$:DC$(I%)=STR$(LB)::DE$(I%)=STR$(P2):DF$(I%)=Q$
5597 PRINT P2,P1
5600 NEXT I%
5601 P1$=STR$(P1)
5605 P1=P1/VAL(Q1$)
5606 P1$=STR$(P1)
5610 IF PO=0 THEN RETURN
```

```
5615 PRINT P1
5616 PRINT DA(2)
5620 FOR I%=1 TO N2%
5625 CN$=STR$(DA(I%)):GET 1,DA(I%),1140
5630 N$=DB$(I%):LB$=DC$(I%):P2$=DE$(I%):Q$=DF$(I%)
5640 PRINT DA(I%),N$,Q$,P2$:  PUT 1,DA(I%),1140
5650   GOSUB 8400:NEXT I%:PRINT P1
5651 CLOSE:RETURN
6000 REM *** SUB HELP ********************************************
6010 PRINT CHR$(31)
6020 PRINT @25,"HELP ROUTINE"
6030 PRINT @64,STRING$(64,"*")
6040 PRINT @ 138,"1. INGREDIENT LIST"
6050 PRINT @ 202,"2. INDIVIDUAL INGREDIENT INFORMATION"
6060 PRINT@ 266,"3.  MAIN MENU
6070 PRINT@330,"4.  RATION LIST NAMES
6080  PRINT @896 ,"ENTER NUMBER ":INPUT O$
6090 IF O$="1" THEN GOSUB 60050
6100 IF O$="2" THEN GOSUB 60300
6105 PP=1
6110 IF O$="3" OR  O$ ="M" THEN RETURN
6111 IF O$="4" THEN GOSUB 6300
6200 GOTO 6000
REM *** Unit name listing printouts ***************************
6300 A=0:WW=1:CLS
6310 LPRINT TAB(38);"UNIT LISTING"
6320 GOSUB 3000
6325 F9=1
6330 LPRINT STRING$(80,"*")
6340 LPRINT :GOTO 6370
6350 GOSUB 3517
6360 F9=F9+1
6370 LPRINT F9;". ";RI$
6380 IF PEEK(16425)>52 THEN LPRINT CHR$(12)
6390 IF EOF(1) THEN WW=0:LPRINT CHR$(12):CLOSE:RETURN
6400 GOTO 6350
6610 Z$=LB$:GOSUB 40000:IF Z=1 THEN 600
REM *** Puchases subroutine **********************************
7000 PRINT @30,"PURCHASES"
7005 PP=0
7010 PRINT @64,STRING$(64,"*")
7020 PRINT @202,"1.  DATE (MO/DA/YR) - ";D2$
7021 T1$=INKEY$:IF T1$="" THEN 7021
7022 IF T1$="Y" THEN GOTO 7040
7029 PRINT @202,"";
7030 LINE INPUT "1.  DATE (MO/DA/YR) - ";D2$
7035 Z$=D2$:GOSUB 45000:IF Z=1 THEN PRINT @200,CHR$(31):GOTO 7020
7040 PRINT @266,"";
7050 LINE INPUT "2.  INGREDIENT # - "; I1$
7055 Z$=I1$:GOSUB 40000:IF Z=1 THEN PRINT @264,CHR$(31):GOTO 7040
7060 PRINT @330,"";
7070 LINE INPUT "3.  QUANITY - ";Q2$
7075 Z$=Q2$:GOSUB 40000:IF Z=1 THEN PRINT @328,CHR$(31):GOTO 7060
```

```
7080 PRINT @394,"";
7090 LINE INPUT "4.  VALUE - ";V1$
7095 Z$=V1$:GOSUB 40000:IF Z=1 THEN PRINT @392,CHR$(31):GOTO 7080
7096 IF VAL(V1$)<=1.00 THEN GOTO 7101:ELSE  PRINT"IS THIS VALUE
     CORRECT (Y OR N)IN PRICE /LB"
7097 T1$=INKEY$:IF T1$="" THEN 7097
7098 IF T1$="N" THEN PRINT @392,CHR$(31):GOTO 7080
7099 IF T1$="Y" THEN GOTO 7101
7100 PRINT @ 448,CHR$(31):GOTO 7096
7101 PRINT "ARE ALL OF THESE CORRECT (Y OR N)"
7102 T1$=INKEY$:IF T1$="" THEN 7102
7103 IF T1$="Y" THEN GOTO 7110:ELSE GOTO 7000
7110 OPEN "R",1,"INV","FF",36
7120 I1=VAL(I1$)
7130 GET 1,I1,1140
7140 P=((VAL(LB$)*VAL(P$))+(VAL(Q2$)*VAL(V1$)))/(VAL(LB$)+VAL(Q2$))
7150 P$=STR$(P)
7160 LB=VAL(LB$)+VAL(Q2$)
7170 LB$=STR$(LB)
7180 PUT 1,$,1140
7190 CLOSE
7195 IF PP=1 THEN RETURN
7200 OPEN "E",1,"PURC:1","FF",50
7201 P3=VAL(Q2$)*VAL(V1$):P3$=STR$(P3)
7205 CR=CR+1
REM *** Stores the purchases ***********************************
7210 PUT 1,,1160
7220 CLOSE
7230 PRINT @896,"ENTER TO CONTINUE OR <M>AIN":INPUT C2$
7240 IF C2$="" THEN CLS:GOTO 7000
7250 IF C2$="M" THEN RETURN
7260 IF C2$="H" OR C2$="HELP" THEN CLS:GOSUB 6000
7270 GOTO 7210
8400 REM *** SUB FOR ING ***************************************
8402 I=1: IF VAL(U1$) = 11 THEN U1$="11" : GOTO 8416
8404 IF VAL(U1$) = 12 THEN U1$="12" : GOTO 8416
8406 IF VAL(U1$) = 13 THEN U1$="13" : GOTO 8416
8408 IF VAL(U1$) = 14 THEN U1$= "14" : GOTO 8416
8410 IF VAL(U1$) = 15 THEN U1$="15" : GOTO 8416
8414 IF VAL(U1$) = 10 THEN U1$="10":GOTO 8416
8415 IF VAL(MID$(U1$,I,1))>=1 AND VAL( MID$(U1$,I,1))<=9 THEN
     U1$="0"+MID$(U1$,I,1):ELSE I=I+1:GOTO 8415
8416 I=1
8417 IF VAL(CN$)>9 THEN X=VAL(CN$):CN$=STR$(X):CN$=MID$(CN$,2,2):GOTO
     8430
8420 IF VAL(MID$(CN$,I,1))>=1 AND VAL(MID$(CN$,I,1) )<=9AND G$="I" THEN
     CN$="0" + MID$(CN$,I,1)ELSE IF I>LEN(CN$) THEN GOTO 8421ELSE
     I=I+1:GOTO 8420
8421 IF VAL(CN$)<10 AND  G$="R" THEN CN$="0"+MID$(STR$(DA(I%)),2,1)
8430 PUT 2,,1170
8440 RETURN
8800 REM ***********MONTH END REPORTS***************
8810 CLS
```

```
8820 PRINT @ 25,"MONTH END REPORTS"
8830 PRINT @64,STRING$(64,"*")
8840 PRINT@202,"1. MONTH END REPORTS
8850 PRINT@266,"2. INGREDIENT LIST
8860 PRINT@330,"3. RETURN TO REPORT MENU
8880 PRINT@704,STRING$(64,"*")
8890 PRINT@778,"ENTER NUMBER";:INPUT A$
8900 Z$=A$:GOSUB 40000
8910 IF Z=1 THEN GOTO 8810
8920 A=VAL(A$):IF A<1 OR A>5 THEN GOTO 8810
8930 IF A=1 THEN GOSUB 16000:RUN"REPORTS"
8940 IF A=2 THENGOSUB 13000
8950 IF A=3 THENRETURN
8970 IF A=4 THEN GOSUB 13500
8980 GOTO 8800
10000 REM *** REPORTS ****************************************
10010 PRINT @30,"REPORT MENU"
10020 PRINT @64,STRING$(64,"*")
10030 PRINT @212,"1. MONTH END REPORT
10050 PRINT @276,"2. INVENTORY LIST
10060 PRINT @340,"3. RATION LIST
10070 PRINT @407,"(RATION CONTENT)
10080 PRINT @468,"4. PURCHASES TO DATE(MONTH)
10090 PRINT @532,"5. SALES TO DATE(MONTH)
10100 PRINT @596,"6. RETURN TO MAIN
10110 PRINT @906,"ENTER NUMBER";
10120 INPUT ET$
10125 IF ET$="6" OR ET$="M" OR ET$="MAIN" THEN RETURN
10130 IF ET$>"6" OR ET$<"1" THEN PRINT @960,"ONLY ENTER NUMBERS
      BETWEEN 1 AND 6 (<H>ELP) PRESS ENTER TO REDO":INPUT
      BY$:CLS:GOTO 10000
10140 ST%=PEEK(14312) AND 240
10142 IF ST%<>48 THEN PRINT@906, "PRINTER UNAVAILABLE PRESS ENTER
      WHEN READY":INPUT ST$:CLS:GOTO 10000
10144 PRINT "PRINTER IS AVAILABLE,PRESS ENTER TO CONTINUE":INPUT
      ST$
10150 ET = VAL(ET$)
10155  POKE 16425,0
10160 ON ET GOSUB 8800,13000,12100,16000,17000
10170 CLS:GOTO 10000
12100 REM *****TOTAL UNIT RATION LISTING**********
12110 A=0:WW=1:CLS
12120 LPRINT TAB(38);"UNIT LISTING"
12130 GOSUB 3000
12135 F9=1
12140 LPRINT STRING$(80,"*")
12150 LPRINT:GOTO 12160
12155 GOSUB 3517
12156 F9=F9+1
12160 LPRINT TAB(10);F9;". RATION ID - ";RI$
12170 LPRINT "NO.      ID#     INGREDIENT         PERCENT "
12180 LPRINT STRING$(80,"-")
12190 AZ$="##       ##       %         %      ###.##"
```

```
12200 TP=0:N1%=1:FOR N%=1 TO N2%:TP=TP+PC(N%)
12205 IF PEEK(16425)>52 THEN LPRINT CHR$(12)
12210 NA$=IV$(IN%(N%)):LPRINT USING AZ$;N%,IN%(N%),NA$,PC(N%)
12220 NEXT N%
12230 LPRINT  TAB(60);"-----"
12240 LPRINT TAB(60);TP
12245 LPRINT:LPRINT STRING$(80,"#")
12246 IF EOF(1) THEN WW=0:LPRINT CHR$(12): RETURN
12250 GOTO 12155
13000 CLOSE:OPEN "R",1,"INV","FF",36
13010 LPRINT TAB(38) "INVENTORY"
13020 JJ$=" ##.      %         %    ######.##     $###.####"
13030 LPRINT
13040 LPRINT STRING$(80,"*")
13045 LPRINT"NUMBER     NAME            QUANITY     VALUE"
13046 LPRINT STRING$(80,"-")
13050 LPRINT
13060 I=0
13070 I=I+1
13080 GET 1,I,1140
13085 LB=VAL(LB$):P=VAL(P$)
13090 LPRINT USING JJ$;I,N$,LB,P
13100 IF EOF(1) THEN CLOSE:LPRINT CHR$(12):RETURN
13105 IF PEEK(16425)>52 THEN LPRINT CHR$(12)
13110 GOTO 13070
15375 STOP
16000 REM *********PURCHASES REPORT*************************
16005 DF = 0 : DE = 0
16010 CLS
16020 LPRINT TAB(38);"PURCHASES"
16030 LPRINT STRING$(80,"*")
16040 LPRINT
16050 LPRINT"   DATE    NO. INGREDIENT      QUANITY    VALUE($/LB)
      TOTAL"
16060 LPRINT STRING$(80,"-")
16070 AX$="### %     %  %  % %       % ########.###   ###.####
      ########.####"
16080 CLOSE:OPEN "R",1,"PURC:1","FF",50
16083 I=0
16085 IF EOF(1) THEN LPRINT CHR$(12): GOTO 16116
16087 I=I+1
16090 GET 1,,1160
16105 Q2=VAL(Q2$):V1=VAL(V1$)
16106 DF=DF+VAL(P3$)
16107 ZZ=VAL(P3$)
16108 DE=DE+Q2
16109 IF PEEK(16425)>52 THEN LPRINT CHR$(12)
16110 LPRINT USING AX$;I,D2$,I1$,IV$(VAL(I1$)),Q2,V1,ZZ
16115 GOTO 16085
16116 LPRINT:LPRINT TAB(35) "TOTAL PURCHASES":LPRINT
      STRING$(80,"*"):LPRINT :LPRINT "TOTAL POUNDS - ";DE:LPRINT "TOTAL
      COST - ";DF:LPRINT CHR$(12)
```

```
16117   DF$=STR$(DF):DE$=STR$(DE):CLOSE:OPEN "R",1,"P:0","FF",20:PUT
      1,1,1186
16120 RETURN
17000 REM *** SALES UP TO DATE FOR MONTH *****************
17010 LPRINT TAB(30);"SALES UP TO DATE FOR THE MONTH"
17015 DE=0
17020 LPRINT STRING$(80,"+")
17030 LPRINT
17040 LPRINT " DATE   UNIT# TYPE ID#  QUANITY  PROJECT#    KIND  $/LB
       TOT/COST"
17050 AV$="%      % ##  !  ## #########.###  %         %!  #.#####
      ########.##"
17060 CLOSE:OPEN "R",1,"SALE:!","FF",51
17070 IF EOF(1) THEN LPRINT CHR$(12):RETURN
17080 GET 1,,1150
17085 U1=VAL(U1$):Q1=VAL(Q1$):B1=VAL(B1$)
17086 DE=DE+(Q1*VAL(P1$))
17087 ZQ=Q1*VAL(P1$)
17088 P1=VAL(P1$)
17090 IF PEEK(16425)>52 THEN LPRINT CHR$(12)
17091 LPRINT USING AV$;D1$,U1,G$,B1,Q1,PJ$,S1$,P1,ZQ
17100 GOTO 17070
31345 STOP
40000 REM ERROR SUB *******************************************
40010 Z=0
40020 X=LEN(Z$)
40025 IF X=1 AND(Z$="H" OR Z$="HELP") THEN POKE 16916,0: CLS:GOSUB
      6000:RETURN
40030 FOR J=1TOX
40035 ZZ$=MID$(Z$,J,1)
40040 IF ZZ$="0" THEN GOTO 40100
40050 IF ZZ$="/" THEN GOTO 40100
40060 IF ZZ$="."THEN GOTO 40100
40063 IF ZZ$="@" THEN GOTO 40100
40065 IF ZZ$=" " AND X>1 THEN 40100
40070 IF VAL(ZZ$) = 0 THEN Z=1:PRINT CHR$(27);CHR$(31)
40080 IF ZZ$="M" AND X=1 THEN Z=2
40090 REM
40100 NEXT J:RETURN
40300 REM ***********YES / NO SUB**************************
40310 Y1=0
40320 IF MID$(Y1$,1,1)="Y" THEN Y1=1 ELSE Y1=2
40330 IF MID$(Y1$,1,1)<>"Y" AND MID$(Y1$,1,1)<>"N" THEN PRINT"ANSWER
      ONLY YES OR NO,REDO":INPUT Y5$
40340 RETURN
45000 REM************************DATE ERROR SUB**********
45005 Z=0
45010 X=LEN(Z$)
45020 IF X<>8 THEN Z=1:RETURN
45030 FOR J=1 TO  X
45035   IF X<>8 AND ZZ$<>" " THEN 45150
45040   ZZ$=MID$(Z$,J,1)
45041 IF J=4 AND VAL(ZZ$)>3 THEN Z=1
```

```
45042 IF J=1 AND VAL(ZZ$)>1, THEN Z=1
45043 IF J=7 AND VAL(ZZ$)<>8 THEN Z=1
45044 IF J=8 AND VAL(ZZ$)<3 THEN Z=1
45045 IF ZZ$=" " THEN 45150
45050   IF ZZ$="0" THEN GOTO 45150
45055 IF (J=3 OR J=6) AND ZZ$<>"/" THEN Z=1
45060   IF ZZ$="/" THEN GOTO 45150
45070   IF VAL(ZZ$)=0 THEN Z=1
45150 NEXT J
45155 RETURN
REM *** Feedmill picture and INV names loaded ****************
50000 CLS
50010 PRINT CHR$(23)
50020 PRINT @16 ,"KANSAS STATE
50030 PRINT@82,"UNIVERSITY
50040 PRINT @210,STRING$(1,151):PRINT @212,STRING$(7,131)
50050 PRINT@224,STRING$(1,171)
50060 PRINT @ 274,STRING$(1,149):PRINT@286,STRING$(6,183)
50070 PRINT@336,STRING$(2,191):PRINT @350,STRING$(6,191)
50080 PRINT @396,STRING$(6,183):PRINT@414,STRING$(6,187)
50090 PRINT @460,STRING$(6,191):PRINT@478,STRING$(6,153)
50100 PRINT @ 524,STRING$(6,153):PRINT@542,STRING$(6,191)
50110 PRINT @588,STRING$(6,187):PRINT@606,STRING$(6,183)
50120 PRINT @652,STRING$(6,191):PRINT@670,STRING$(6,187)
50130 PRINT@716,STRING$(6,183):PRINT@734,STRING$(6,183)
50140 PRINT@780,STRING$(6,183):PRINT@798,STRING$(6,187)
50150 PRINT@916,"FEEDMILL";
50151 I=0
50152 CLOSE:OPEN "R",1,"INV","FF",36:
50153 IF EOF(1) THEN GOTO 50160
50154 I=I+1:GET 1,I,1140
50155 IV$(I)=N$
50156 IF EOF(1) THEN GOTO 50160
50157 GOTO 50153
50160 FOR J = 1 TO 200:NEXT J
50170 RETURN
REM *** Sales corrections ************************************
51000 CLOSE:CLS
51002 PO=1
51005 POKE 16916,3
51010 PRINT @15,"SALES CORRECTIONS
51020 PRINT @64,STRING$(64,"*")
51030 OPEN "R",3,"SALE:1","FF",51
51040 PRINT @132,CHR$(31);:PRINT@132,"WHICH SALE NEEDS CORRECTED OR
      ADJUSTED";:INPUT IA$:Z$=IA$:GOSUB 40000:IF Z=1 THEN GOTO 51040
51060 GET 3,VAL(IA$),1150
51063 GOTO 51070
51064  U1$=UU$:G$=GG$:P1$=PP$:Q1$=QQ$:S1$=SS$:B1$=BB$
51065 POKE 16916,2:CLS
51070 CLS:PRINT@202,"1. DATE - ";D1$
51080 PRINT@266,"2. UNIT #";U1$:UU$=U1$
51090 PRINT@ 330,"3. TYPE (R OR I) - ";G$:GG$=G$
51100 PRINT@394,"4. ID# - "; B1$:BB$=B1$
```

```
51110 PRINT@ 458,"5. QUANITY - ";Q1$:QQ$=Q1$
51120 PRINT@ 522,"6. PROJECT # - ";PJ$
51130 PRINT@586,"7. <B>ULK OR <S>ACK - ";S1$:SS$=S1$
51140 PRINT @650,"8. TOTAL PRICE -";VAL(P1$)*VAL(Q1$)
51150 PRINT@714,"9. PRICE/LB$ - ";P1$:PP$=P1$
51155 P4=VAL(P1$)*VAL(Q1$):P4$=STR$(P4)
51160 PRINT @770,"ARE THESE CORRECT(Y OR N)?";:INPUT B$:Y1$=B$:GOSUB
      40300:IF Y1=1 THEN PUT 3,$,1150:RETURN
51170 IF B$="N" OR B$="NO" THEN GOTO 51180:ELSE PRINT
      @770,CHR$(31);:GOTO 51160
51180  PRINT@770,CHR$(31);:PRINT @770,"WHAT NUMBER NEEDS
      CHANGED?";:INPUT A1$:Z$=A1$:GOSUB 40000:IF Z=1 THEN PRINT
      @770,CHR$(31);:GOTO 51180
51190 PRINT @714,CHR$(31)
51200 A1=VAL(A1$)
51210 ON A1 GOTO 51220,51230,51240,51250,51260,51270,51280,51290,51300
51220 PRINT @778,"1. CORRECTED DATE - ";:INPUT D1$
51225 GOTO 51301
51230 PRINT @778,"2. CORRECTED UNIT - ";:INPUT U5$
51235 GOTO 51301
51240 PRINT @778,"3. <R>ATION OR <I>NGREDIENT - ";:INPUT G5$
51245 GOTO 51301
51250 PRINT @778,"4. CORRECTED ID# - ";:INPUT B5$
51255 GOTO 51301
51260 PRINT @778,"5. CORRECTED QUANITY - ";:INPUT Q5$
51265 GOTO 51301
51270 PRINT @778,"6. PROJECT # -";:INPUTPJ$
51275 GOTO 51301
51280 PRINT @778,"7. <B>ULK OR <S>ACK - ";:INPUT S2$
51285 GOTO 51301
51290 PRINT @778,"8. TOTAL PRICE - ";:INPUTP4$
51295 GOTO 51301
51300 PRINT @778,"9. CORRECTED PRICE/LB - ";:INPUTP1$
51301 FG=1
51305 IF A1$="1" OR A1$="6" THEN GOTO 51465
REM *** Zeros out incorrect sale ******************************
51306 IF VAL(A1$)>1 AND VAL(A1$)<7 THEN GOSUB 5505:GOTO 51456
51310  IF A1$="8" THENP1=VAL(P4$)/VAL(Q1$):P1$=STR$(P1):P3$=P1$:GOTO
      51463
51311 IF A1$="9" THEN P3$=P1$:GOTO 51463
51312 IF A1$="7" THEN GOTO 51313 ELSE 51320
51313 IF S1$="B "  OR S1$="B"THEN P1=VAL(P1$)/1.12
51314 IF S1$="S " THEN P1=VAL(P1$)/1.18
51315 IF S2$="S" THEN P1=P1*1.18
51316 IF S2$="B" THEN P1=P1*1.12
51317 P1$=STR$(P1):S1$=S2$
51318 GOTO 51465
51320 GOTO 51456
51456 IF A1$="2" THEN U1$=U5$
51457 IF A1$="3" THEN G$=G5$
51458 IF A1$="4" THEN B1$=B5$
51459 IF A1$="5" THEN Q1$=Q5$
51461 GOSUB 5500:IF PO=0 THEN GOTO 51064
```

```
51463 IF A1$="9" OR A1$="8" THEN P1$=P3$
REM *** Resaves the correct sale ****************************
51465  CLOSE 3:OPEN "R",3,"SALE:1","FF",51
51470 PUT 3,VAL(IA$),1150
51475 GOTO 51070
51480 RETURN
60030 REM *** INGREDIENT LIST SUB ******************************
60040 L=0
60050 OPEN"R",1,"INV","FF",36
60051 CLS
60052 I=0
60053 POKE 16916,16
60055 D=0
60057 F=64
60080 I = I + 1
60110 GET 1,I,1140
60120 N1$=N$
60170 D=D+F
60200 PRINT@D,I;"";N1$;
60205 IF I =14 OR I= 56 OR I=98 THEN D=20
60210 IF I = 28 OR I=70  OR I=112 THEN D=40
60220 IF I =42 OR I = 84 THEN PRINT @960,"PRESS ANY KEY TO
      CONTINUE":POKE 16916,0:INPUT O$:CLS:GOTO60053
60230 IF EOF(1) THEN CLOSE :GOTO 60250
60240 GOTO 60080
60250 PRINT @960,"DO YOU WANT TO SEE THE INGREDIENTS AGAIN":INPUT
      O$
60260 IF O$="YES" OR O$ = "Y" THEN CLOSE:GOTO 60050
60270 CLS:RETURN
60280 GOTO  60240
60300  REM *** SUB FOR INDIVIDUAL INGREDIENTS **********************
60304 CLS
60305 POKE 16916,20
60310 PRINT @20,"INDIVIDUAL INGREDIENTS
60320 PRINT@64,STRING$(64,"*")
60340 PRINT @896 ,"WHICH INGREDIENT NUMBER DO YOU WANT TO LOOK
      AT?"
60350 INPUT O$
60355 OPEN "R",1,"INV","FF",36
60360 O=VAL(O$)
60370 GET 1,O,1140
60380 PRINT @266,"INGREDIENT NUMBER - ";CN$
60390 PRINT @330,"INGREDIENT - ";N$
60395 PRINT @394,"QUANITY - ";LB$
60400 PRINT @458,"VALUE - ";P$
60410 PRINT @896,"PRESS ENTER TO  RETURN TO HELP MAIN":INPUT
      O$:POKE 16916,0:CLS:RETURN
REM *** Purchase corrections and adjustments *********************
61000 CLS:PRINT @15,"PURCHASE CORRECTIONS AND ADJUSTMENTS"
61010 CLOSE
61020 OPEN  "R",1,"PURC:1","FF",50
61030 PRINT @ 64,STRING$(64,"*")
```

```
61040 PRINT @132,"WHICH PURCHASE NEEDS ADJUSTED OR
      CORRECTED";:INPUT A$:AA=VAL(A$)
61050 IF A$ ="M" THEN PRINT"STOP
61060 IF VAL(A$)<1 OR VAL(A$)>LOF(1) THEN CLS:CLOSE:GOTO 61000
61070 GET 1,VAL(A$),1160
61080 PRINT @ 132,CHR$(31):PRINT @132, "1. DATE - ";D2$
61090 PRINT@196,"2. INGREDIENT # - ";I1$
61100 PRINT@260,"3. QUANITY - ";Q2$
61110 PRINT@324,"4. VALUE - ";V1$
61120 PRINT@388,"5. TOTAL PRICE - ";P3$
61130 IF FD =1 THEN GOTO 61340
61140 PRINT@450,"ARE THESE CORRECT (Y OR N)";:INPUT A$
61150 IF MID$(A$,1,1)="Y" THEN CLOSE: OPEN "R",1,"PURC:1","FF",50: PUT
      1,60,1160:CLOSE:GOSUB 7110:RETURN
61160 IF MID$(A$,1,1)<>"N" THEN GOTO 61080
61170 CLOSE:OPEN "R",2,"INV","FF",36
61180 GET 2,VAL(I1$),1140
61190 P=(VAL(LB$)*VAL(P$)-(VAL(Q2$)*VAL(V1$)))/(VAL(LB$)-VAL(Q2$))
61200 P$=STR$(P)
61210 LB=(VAL(LB$)-VAL(Q2$))
61220 LB$=STR$(LB)
61230 PUT 2,VAL(I1$),1140
61240 PRINT @450,CHR$(31):PRINT @450,"ENTER NUMBER TO BE
      CORRECTED";:INPUT A$
61250 A=VAL(A$):IF A<1 OR A>5 THEN GOTO 61240
61260  REM IF A=5 THEN PRINT@450, CHR$(31):PRINT @450, "FOR TOTAL
      ADJUSTMENTS GOTO MENU FOR ADJUSTMENTS <ENTER> ";:INPUT
      B$:GOTO 61000
61270 IF A=1 THEN LINE INPUT "1. DATE - ";D2$:Z$=D2$:
61280 IF A=2 THEN LINE INPUT "2. INGREDIENT # - ";I1$
61290 IF A=3 THEN LINE INPUT "3. QUANITY - ";Q2$
61300 IF A=4 THEN LINE INPUT "4. VALUE - ";V1$
61305 IF A=5 THEN LINE INPUT "5. TOTAL - ";P3$
61310 GOTO 61080
62000 REM ***********MENU FOR CORRECTION ********************
62005 CLS
62010 PRINT@15,"MENU FOR CORRECTIONS AND ADJUSTMENTS
62020 PRINT@64,STRING$(64,"*")
62030 PRINT@148,"1. PURCHASE CORRECTIONS
62040 PRINT@212,"2. PURCHASE PRICE ADJUSTMENTS
62050 PRINT@276,"3. SALES CORRECTION
62060 PRINT@340,"4. MAIN MENU
62080 PRINT@468," ENTER SELECTIONS";:INPUT A$
62090 A=VAL(A$):IF A<1 OR A>4 THEN CLS:GOTO 62010
62100 IF A=1 THEN FD=0:GOSUB 61000:GOTO 62000
62110 IF A=2 THEN FD=1:GOSUB 61000:GOTO 62000
62120 IF A=3 THEN ED=0:GOSUB 51000:GOTO 62000
62140 IF A=4 THEN RETURN
REM *** INVENTORY DUPLICATION CHECK **********************
63000  M=0
63010 FOR I=1 TO X9
63020 IF N$=I1$ THEN PRINT N$;" IS ALREADY IN INVENTORY AT INGREDIENT
      # ";I:M=1:INPUT M$
```

```
63030 NEXT I:RETURN
REM *** ALPHABETIZED INVENTORY LIST **********************
63500 DIM A%(125)
63510 CMD"O",X9,*A%(1),IV$(1)
63520 FOR I=1 TO X9
63530 LPRIN TA%(I);IV$(A%(I))
63540 NEXT I
63550 RETURN
65000 CLOSE:RESUME 10
```

APPENDIX B.2

REPORTS PROGRAM LISTING

```
1 REM *** REPORTS ************************************
2 :CLS:KK=0
3 PRINT@20,"MONTH END REPORTS":CLEAR 5000
4 PRINT @138,"HOW MANY COPIES DO YOU WANT";:INPUT A$
5 X=VAL(A$)
6 IF X<1 OR X>10 THEN GOTO 2
8 CLS:PRINT @20,"MONTH END REPORTS":
13 XX=0
REM *** PRINT OPTIONS ********************************************
14 CLS:PRINT @74,"1. MONTH TO DATE REPORTS
15 PRINT @138,"2. MONTH END CLOSEOUTS
16 PRINT @202,"3. RETURN TO MAIN MENU
17 PRINT @266,"ENTER NUMBER";:INPUT A$:ZZ$=A$
18 IF VAL(A$)<1 OR VAL(A$)>4 THEN GOTO 15
19 IF A$="1" THEN XX=0 :GOTO 24
20 IF A$="2" THEN IF KK=0 THEN XX=1:GOTO 24
21 IF A$="3" THEN RUN "MAIN/COM" ELSE IF A$="4" THEN GOSUB 4500
22 GOTO 15
23 POKE 16425,0
REM *** MAKING SURE THERE IS A BACKUP ***********************
24  CLS:PRINT "HAVE YOU MADE A BACKUP DISK FOR THE DAY YET? IF
      YOU HAVE NOT  * STEP ONE : GO BACK TO MAIN MENU * STEP TWO :
      HIT NUMBER 8 *STEP THREE : ACT AS IF YOU ARE DONE FOR THE
      DAY * STEP FOUR  : AFTER YOU HAVE BACKED UP BOTH DISKS AND
      YOU RECEIVE ";
25 PRINT "A THE END PROMPT REBOOT THE MACHINE(HIT THE ORANGE
      BUTTON IN UPPER RIGHT HAND  CORNER OF KEY BOARD *STEP FIVE :
      WHEN MAIN MENU REAPPERS GO  BACK TO REPORTS MENU AND
      CONTINUE AS BEFORE!!! (HIT <C> TO    CONTINUE,<M>FOR MAIN
      MENU";:INPUT A$
26  IF A$="M" THEN A$="3":GOTO 21 : ELSE IF A$="C" THEN GOTO 27 ELSE
      GOTO 2
27 IF KK=1 THEN CLEAR 1000
28 CLS:PRINT CHR$(23):PRINT @18,"REPORTS
30 AI=1:AJ=1:CT=0:TC=0:JJ=0
40 XB$="###  %       % ########.## ##.#### #########.### #########.##
      #########.####"
50 DIM K1%(515),K3%(515),K4(515),K5(515),A%(515)
60 DIM KA%(300),KC%(300),KD(300),KE(300),B%(300)
70 DIM AK%(515),CK%(515),DK(515),EK(515),FK(515),GK(515)
REM *** INITIALIZES TOT AND ING ARRAYS ***********************
80 CLOSE:OPEN "R",1,"TOT:0","FF",42
90 IF EOF(1) THEN GOTO 160
100 TC=TC+1
110 GET 1,TC,10010
140 KA%(TC)=VAL(U1$)::KC%(TC)=VAL(CN$):KD(TC)=VAL(TG$):KE(TC)=VAL(PT$)
150 GOTO 90
160 CLOSE:OPEN "R",1,"ING:1","FF",44
170 IF EOF(1) THEN GOTO 240
```

```
180 CT=CT + 1
190 GET 1,CT,10000
220 K1%(CT)=VAL(U1$):K3%(CT)=VAL(CN$):K4(CT)=VAL(Q$):K5(CT)=VAL(P2$)
230 GOTO 170
REM *** SORTS THE ARRAYS ****************************************
240 IF CT=0 THEN GOTO 260
250 CMD"O",CT,*A%(1),K1%(1),K3%(1)
260 IF TC=0 THEN GOTO 280
270 CMD"O",TC,*B%(1),KA%(1),KC%(1)
280 IF AI>CT AND AJ>TC THEN
      CMD"F=ERASE",K1%(515),K3%(515),K4(515),K5(515): GOSUB 1000: LPRINT
      CHR$(12):GOTO 500
290 P=A%(AI)
300 IF TC<>0 THEN Q=B%(AJ):
305 JJ=JJ+1
306 IF TC=0 THEN GOTO 410
REM *** COMPARES UNIT NUMBER ************************************
310 IF K1%(P)=KA%(Q) THEN GOTO 340
312 IF KA%(Q)=0 THEN 410
314 IF K1%(P)=0 THEN 380
320 IF K1%(P)<KA%(Q)  THENGOTO 410
330 IF K1%(P)>KA%(Q)  THEN GOTO 380
340 IF K3%(P)=KC%(Q) THEN GOTO 350 ELSE GOTO 370
350 AK%(JJ)=K1%(P):CK%(JJ)=K3%(P):DK(JJ)=K4(P):EK(JJ)=K5(P):FK(JJ)=K4(P)+
      KD(Q) :GK(JJ)=(K4(P)*K5(P)+KD(Q)*KE(Q))/FK(JJ)
360 AI=AI+1:AJ=AJ+1:GOTO 280
REM *** COMPARES INGREDIENT NUMBERS ******************************
370 IF K3%(P)>KC%(Q) THEN GOTO 380 ELSE GOTO 400
380 AK%(JJ)=KA%(Q):CK%(JJ)=KC%(Q):DK(JJ)=0:EK(JJ)=0:FK(JJ)=KD(Q):GK(JJ)
      =KE(Q )
390 AJ=AJ+1:GOTO 280
400 IF K3%(P)<KC%(Q) THEN GOTO 410 ELSE PRINT "ERROR 1
410 AK%(JJ)=K1%(P):CK%(JJ)=K3%(P):DK(JJ)=K4(P):EK(JJ)=K5(P):FK(JJ)=K4(P):
      GK(JJ )=K5(P)
420 AI=AI+1:GOTO 280
500 REM  *** REPORTS FOR PROJECTS AND RATIONS***************
501 CMD"F=KEEP",X,XX,H6
502 DIM S1%(150),S2$(150),S3(150),S4(150),S5$(150),S6%(150),C%(150)
503 DIM SA%(150),SB$(150),SC(150),SD(150),D%(150)
504 DIM MA%(250),MB$(250),MC(250),MD(250),ME(250),MF(250)
505 DIM AS%(250),BS$(250),CS%(250),DS(250),ES(250),E%(250)
506 DIM F1%(250),F2$(250),F3%(250),F4(250),F5(250),F6(250),F7(250)
507 XC$="%          % #########.### ##.#### #########.### #########.##
      #########.####"
508 XD$="###      #########.## ##.####  #########.###  #########.##
      #########.####"
509 CLS:POKE 16916,1:PRINT @30,"PROJECTS REPORTS":PRINT @66,"PLEASE
      ADJUST PAPER ON THE PRINTER TO THE TOP OF THE PAGE ONCE
      AGAIN (HIT ENTER TO CONTINUE)";:INPUT A$
REM *** INITIALIZES SALE ARRAYS **********************************
510 CLOSE:OPEN "R",1,"SALE:1","FF",51
520 IF EOF(1) THEN GOTO 590
525 CLS:PRINT@126,"*"
```

```
530 BT=BT+1
540 GET 1,,10020
560 S1%(BT)=VAL(U1$):S2$(BT)=PJ$:S3(BT)=VAL(Q1$):S4(BT)=VAL(P1$):S5$(BT)=
    G$:S6 %(BT)=VAL(B1$)
570 GOTO 520
REM *** INITIALIZES TPRO ARRAYS **********************************
590  CLOSE:OPEN "R",1,"TPROJ:0","FF",36
600 IF EOF(1) THEN GOTO 660
605 CLS:PRINT @126,"*"
610 TT=TT+1
620 GET 1,TT,10030
640 SA%(TT)=VAL(U1$):SB$(TT)=PJ$:SC(TT)=VAL(QQ$):SD(TT)=VAL(P5$)
650 GOTO 600
REM *** INITIALIZES TRAT ARRAYS **********************************
660 CLOSE:OPEN "R",1,"TRAT:0","FF",40
670 IF EOF(1) THEN GOTO 730
675 CLS:PRINT @ 126,"*"
680 AT=AT+1
690 GET 1,AT,10040
710 AS%(AT)=VAL(U1$):BS%(AT)=VAL(RN$):CS$(AT)=PJ$:DS(AT)=VAL(QQ$):ES(
    AT)=V AL(PP$)
720 GOTO 670
REM *** SORTS THE ARRAYS *****************************************
730 CMD"O",BT,*C%(1),S1%(1),S2$(1)
735 IF TT=0 THEN 745
740 CMD"O",TT,*D%(1),SA%(1),SB$(1)
745 IF AT=0 THEN 760
750 CMD"O",AT,*E%(1),AS%(1),BS%(1)
760   XX=1:GOTO 2000
1000 REM *** PRINT OUTS FOR ING.**********************************
1001 DIM X1%(200),X3(200),X4(200),X5%(200)
1006 U=1
1007 H5=0:H6=0:H7=0:H8=0
1010 FOR D=1 TO X
1012 POKE 16425,0
1020   CO=1
REM *** PRINTS BY UNIT *******************************************
1030   FOR J%=1 TO 13
1031 MM=0
1032 H1=0:H2=0:H3=0:H4=0
1035 IF CO=1 THEN DK(0)=FK(CO)
1040 H1=0:H2=0:H3=0:H4=0: LPRINT TAB(30) "INDIVIDUAL INGREDIENT
    LIST":LPRINT TAB(30) STRING$(25,"-"):LPRINT :GOSUB 3000
1050 LPRINT" NUM      NAME      LBS/MO  $/LB     $/MO      LB/YR
    $/YR":LPRINT STRING$(80,"*"):LPRINT
1051 IF CT=0 THEN DK(CO)=0
1052 IF AK%(CO)=J% THEN MM=1:GOTO1064 ELSE 1070
1057 IF CT=0 THEN DK(CO)=0
REM *** ADDING THE TOTALS ****************************************
1058 H5=H5+DK(CO):H6=H6+DK(CO) *EK(CO):H7=H7+FK(CO):H8=H8+FK(CO)*GK(C
    O) :H1 =H1+DK(CO):H2=H2+DK(CO)*EK(CO):H3=H3+FK(CO)
    :H4=H4+FK(CO)*GK(CO) :
1059 IF PEEK(16425) > 52 THEN LPRINT CHR$(12)
```

```
1060 X1%(U)=AK%(CO):X3(U)=FK(CO):X4(U)=GK(CO):X5%(U)=CK%(CO):U=U+1:
     LPRINT USING XB$;CK%(CO),IV$(CK%(CO)) , DK(CO),EK(CO),
     DK(CO)*EK(CO), FK(CO),GK(C O)*FK (CO):CO=CO+1:GOTO 1051
1064 IF (DK(CO+1)<0 ORDK(CO)<0 OR DK(CO)=0) AND CK%(CO)=CK%(CO+1)
     AND AK%(CO)=AK%(CO+1)  THEN CO=CO+1:DK(CO)=DK(CO-1)+DK(CO):IF
     FK(CO-1)+FK(CO)=0 THEN FK(CO)=0:GOTO 1051 ELSE GOTO 1051
1065 IF  CK%(CO)=CK%(CO+1) AND AK%(CO)=AK%(CO+1) AND DK(CO)<>0THEN
     CO=CO+1:FK(CO)=DK(CO)+FK(CO-1):GK(CO)=(GK(CO-1)*FK(CO-1)+DK(CO)
     *EK(
     CO))/FK(CO):EK(CO)=(EK(CO)*DK(CO)+EK(CO-1)*DK(CO-1))/(DK(CO)+DK(C
     O-1)): DK(CO)=DK(CO)+DK(CO-1):GOTO 1051:ELSE GOTO 1058
1070  IF MM=1 THEN LPRINT:LPRINT STRING$(80,"-"):LPRINT TAB(35)
     "TOTALS":LPRINT TAB(35) STRING$(6,"*"):LPRINT:LPRINT TAB(30)
     "LBS/MO - ";H1:LPRINT TAB(30) "$/MO   - ";H2:LPRINT TAB(30) "LBS/YR
     - ";H3:LPRINT TAB(30)"$/YR   - ";H4:LPRINT CHR$(12)
1080 REM PRINT PAGE
1090    NEXT J%
1105  LPRINT CHR$(12):LPRINT TAB(30)"TOTALS FOR INGREDIENTS":LPRINT
     STRING$(80,"*"):LPRINT:LPRINT TAB(30)"TOTAL POUNDS/MONTH -
     ";H5:LPRINT TAB(30) "TOTAL DOLLARS/MONTH - ";H6:LPRINT TAB(30)
     "TOTAL POUNDS/YEAR - ";H7:LPRINT TAB(30) "TOTAL DOLLARS/YEAR
     - ";H8:
1106 LPRINT CHR$(12):NEXT D
1110 IF XX=1 AND ZZ$="2"THEN GOSUB 4000
1120 RETURN
1200 REM *** PRINTOUTS FOR PROJECTS ******************************
1220 FOR D=1 TO X:CO=1
1221 H1=0:H2=0:H3=0:H4=0
1222 POKE 16425,0
1225 U=1
REM *** PRINT BY UNITS ****************************************
1230   FOR J%=1 TO 13
1231 H1=0:H2=0:H3=0:H4=0
1232 CLS:PRINT @126,"**"
1235 MM=0
1240 GOSUB 3000
1245 LPRINT "PROJECT#          LB/MO   $/LB    $/MO    LB/YR
     $/YR":LPRINT STRING$(80,"*"):LPRINT
1260    IF MA%(CO)=J% THEN MM=1:GOTO  1265 ELSE 1285
1265 IF MC(CO)=0 AND MB$(CO)=MB$(CO+1) AND MA%(CO)=MA%(CO+1) THEN
     CO=CO+1:GOTO 1260
1270 IF MB$(CO)=MB$(CO+1) AND MA%(CO)=MA%(CO+1) THEN
     CO=CO+1:ME(CO)=MC(CO)+ME(CO-1):MF(CO)=(ME(CO-1)*MF(CO-1)+MC(C
     O)*MD
     (CO))/ME(CO):MD(CO)=(MD(CO)*MC(CO)+MD(CO-1)*MC(CO-1))/(MC(CO)+M
     C(CO- 1)):MC(CO)=MC(CO)+MC(CO-1):GOTO 1260:ELSE 1278
1278 IF PEEK(16425) > 52 THEN LPRINT CHR$(12)
1280 H1=MC(CO)+H1:H2=MD(CO)*MC(CO)+H2:H3=ME(CO)+H3:H4=MF(CO)*ME(CO)
     +H4:M
     A%(U)=MA%(CO):MB$(U)=MB$(CO):ME(U)=ME(CO):MF(U)=MF(CO):U=U+1:LP
     RINT USING
     XC$;MB$(CO),MC(CO),MD(CO),MC(CO)*MD(CO),ME(CO),MF(CO)*ME(CO):C
     O=CO +1:GOTO 1260
```

```
1285  IF MM=1 THEN LPRINT:LPRINT STRING$(80,"-"):LPRINT TAB(35)
      "TOTALS":LPRINT TAB(35) STRING$(6,"*"):LPRINT:LPRINT TAB(30)
      "LBS/MO - ";H1:LPRINT TAB(30) "$/MO   - ";H2:LPRINT TAB(30) "LBS/YR
      - ";H3:LPRINT TAB(30)"$/YR   - ";H4:LPRINT CHR$(12)
1290    NEXT J%
1300 NEXT D
1310 IF XX=1 AND ZZ$="2"THEN GOSUB 4100
1320 RETURN
1400 REM *** PRINTOUTS FOR RAITONS ****************************
1410 FOR D=1 TO X:CO=1
1411 H1=0:H2=0:H3=0:H4=0
1412 POKE 16425,0
1415 U=1
REM *** PRINT BY UNITS ****************************************
1420   FOR J%= 1 TO 13
1421 H1=0:H2=0:H3=0:H4=0
1425 MM=0
1430 GOSUB 3000
1435 LPRINT "RATION #     LB/MO    $/LB     $/MO     LB/YR
      $/YR":LPRINT STRING$(80,"*"):LPRINT
1445 IF F1%(CO)=0 THEN 1480
1450   IF F1%(CO)=J% THEN MM=1:GOTO 1455 ELSE 1475
1455 IF F3%(CO)=F3%(CO+1) AND F1%(CO)=F1%(CO+1) AND F4(CO)=0 THEN
      CO=CO+1:GOTO 1450
1460 IF F3%(CO) =F3%(CO+1) AND F1%(CO)=F1%(CO+1) THEN
      CO=CO+1:F6(CO)=F6(CO-1)+F4(CO):F7(CO)=(F6(CO-1)*F7(CO-1)+F4(CO)*F5
      (CO))/
      F6(CO):F5(CO)=(F5(CO)*F4(CO)+F5(CO-1)*F4(CO-1))/(F4(CO)+F4(CO-1)):F4(
      CO)=F 4(CO)+F4(CO-1)::GOTO 1450:ELSE 1468
1468 IF PEEK(16425) > 52 THEN LPRINT CHR$(12)
1470 H1=F4(CO)+H1:H2=F4(CO)*F5(CO)+H2:H3=F6(CO):H4=F6(CO)*F7(CO)+H4:F1
      %(U)=F
      1%(CO):F2$(U)=F2$(CO):F3%(U)=F3%(CO):F6(U)=F6(CO):F7(U)=F7(CO):U=U+1:
      LPRINT USING XD$;F3%(CO),F4(CO),F5(CO),F5(CO)*F4(CO),
      F6(CO),F7(CO)*F6(CO):CO=CO +1:GO TO 1450
1475  IF MM=1 THEN LPRINT:LPRINT STRING$(80,"-"):LPRINT TAB(35)
      "TOTALS":LPRINT TAB(35) STRING$(6,"*"):LPRINT:LPRINT TAB(30)
      "LBS/MO - ";H1:LPRINT TAB(30) "$/MO   - ";H2:LPRINT TAB(30) "LBS/YR
      - ";H3:LPRINT TAB(30)"$/YR   - ";H4:LPRINT CHR$(12)
1480 NEXT J%
1490 NEXT D
1500 IF XX=1 AND ZZ$="2"THEN GOSUB4300
1510 RETURN
2000 REM *** FIGURES PROJECTS REPORTS *************************
2010 AK=1:AL=1:II=0
2020 IF AK>BT AND AL>TT THEN GOSUB 1200:GOTO 2200
2025 II=II+1
2030 CC=C%(AK):IF TT<>0 THEN DD=D%(AL)
2032 IF TT=0 THEN GOTO 2130
2036 IF SA%(DD)=0 THEN GOTO 2130
2037 IF S1%(CC)=0 THEN GOTO 2110
2040 IF S1%(CC)=SA%(DD) THEN GOTO 2090
2050 IF S1%(CC)<SA%(DD)  THEN GOTO 2130
```

```
2060 IF S1%(CC)>SA%(DD)  THEN GOTO 2110
2090 IF S2$(CC)=SB$(DD) THEN
     MA%(II)=S1%(CC):MB$(II)=S2$(CC):MC(II)=S3(CC):MD(II)=S4(CC):ME(II)=S3(C
     C)+SC(
     DD):MF(II)=(S3(CC)*S4(CC)+SC(DD)*SD(DD))/ME(II):AK=AK+1:AL=AL+1:GOT
     O 2020
2100 IF S2$(CC)>SB$(DD) THEN GOTO 2110 ELSE GOTO 2120
2110
     MA%(II)=SA%(DD):MB$(II)=SB$(DD):MC(II)=0:MD(II)=0:ME(II)=SC(DD):MF(II)=SD
     (DD): AL=AL+1:GOTO 2020
2120 IF S2$(CC)<SB$(DD) THEN GOTO 2130 ELSE 2140
2130 MA%(II) =
     S1%(CC):MB$(II)=S2$(CC):MC(II)=S3(CC):MD(II)=S4(CC):ME(II)=S3(CC):MF(II)
     =S4(C C):AK=AK+1:GOTO 2020
2140 PRINT "ERROR
2200 REM *** FIGURES RATIONS REPORTS ****************************
2210 AM=1:AN=1:LL=0
2220 PRINT F7(LL):IF AM>BT AND AN>AT THEN GOSUB 1400:KK=1:LPRINT
     CHR$(12):GOTO 14
2230 IF S5$(C%(AM))="I" THEN AM=AM+1:GOTO 2220
2240 C1=C%(AM):E1=E%(AN)
2245 LL=LL+1
2246 IF AS%(E1)=0 THEN GOTO 2320
2247 IF S1%(C1)=0 THEN GOTO 2300
2250 IF S1%(C1)=AS%(E1) THEN GOTO 2280
2260 IF S1%(C1)<AS%(E1)   THEN GOTO 2320
2270 IF S1%(C1)>AS%(E1)  THEN GOTO 2300
2280 IF S6%(C1)=CS%(E1) THEN
     F1%(LL)=S1%(C1):F2$(LL)=S2$(C1):F3%(LL)=S6%(C1):F4(LL)=S3(C1):F5(LL)=S
     4(C1): F6(LL)=S3(C1)+DS(E1): F7(LL)=(S3(C1)*S4(C1)+DS(E1)*ES(E1))/F6(LL):
     AM=AM +1:AN =AN+1:GOTO 2220
2290 IF S6%(C1)>CS%(E1) THEN GOTO 2300 ELSE 2310
2300 F1%(LL)=AS%(E1):F2$(LL)=BS$(E1):F3%(LL)=CS%(E1):F4(LL)=0:F5(LL)=0:F6(
     LL)=DS (E1):F7(LL)=ES(E1):AN=AN+1:GOTO 2220
2310 IF S6%(C1)<CS%(E1) THEN GOTO 2320 ELSE PRINT "ERROR 2":GOTO 2220
2320 F1%(LL)=S1%(C1):F2$(LL)=S2$(C1):F3%(LL)=S6%(C1):F4(LL)=S3(C1):F5(LL)=S
     4(C1): F6(LL)=S3(C1):F7(LL)=S4(C1):AM=AM+1:GOTO 2220
3000 REM *** SUB FOR TITLES ON THE REPORTS ***********************
3010 LPRINT
3020 ON J% GOTO
     3030,3040,3050,3060,3070,3080,3090,3100,3110,3120,3130,3140,3150
3030 LPRINT TAB(20),"UNIT #1 - HORSE UNIT":RETURN
3040 LPRINT TAB(20),"UNIT #2 - SHEEP UNIT":RETURN
3050 LPRINT TAB(20),"UNIT #3 - SWINE UNIT":RETURN
3060 LPRINT TAB(20),"UNIT #4 - PUREBRED UNIT":RETURN
3070 LPRINT TAB(20),"UNIT #5 - BEEF RESEARCH":RETURN
3080 LPRINT TAB(20),"UNIT #6 - GRASS UNIT":RETURN
3090 LPRINT TAB(20),"UNIT #7 - SHORTHORN UNIT":RETURN
3100 LPRINT TAB(20),"UNIT #8 - COW-CALF UNIT":RETURN
3110 LPRINT TAB(20),"UNIT #9 - DAIRY UNIT":RETURN
3120 LPRINT TAB(20),"UNIT #10 - POULTRY UNIT":RETURN
3130 LPRINT TAB(20),"UNIT #11 - CASSODAY COWS":RETURN
3140 LPRINT TAB(20),"UNIT #12 - KONZA COWS":RETURN
```

```
3150 LPRINT TAB(20),"UNIT #13 - BARTLEY'S COWS":RETURN
3160 LPRINT TAB(20),"UNIT #14 - SPARE 1":RETURN
3170 LPRINT TAB(20),"UNIT #15 - SPARE 2":RETURN
4000 REM **** CLOSE OUTS FOR ING ******************************
4005 CLS:PRINT @ 20 ,"CLOSEOUTS FOR INGREDIENTS":
4010 CLOSE:OPEN "R",1,"TOT:0","FF",42
4020 FOR Z=1 TO
     U-1:U1$=STR$(X1%(Z)):CN$=STR$(X5%(Z)):N$=X2$(Z):TG$=STR$(X3(Z)):PT$=
     STR$( X4(Z)):PUT 1,Z,10010:PRINT U1$,N$,TG$,PT$:NEXT Z:CMD"KILL
     ING:1":RETURN
4100 REM *** CLOSE OUTS FOR PROJECTS ******************************
4101 CLS:PRINT @20,"CLOSE OUTS FOR PROJECTS":PRINT
     @64,STRING$(64,"*")
4105 CLOSE:OPEN "R",1,"TPROJ:0","FF",36
4110 FOR Z=1 TO
     U-1:U1$=STR$(MA%(Z)):PJ$=MB$(Z):QQ$=STR$(ME(Z)):P5$=STR$(MF(Z)):PU
     T 1,Z,10030:PRINT U1$,PJ$,QQ$,P5$:NEXT Z:RETURN
4300 REM *** CLOSEOUTS FOR RATIONS ****************************
4310 CLOSE:OPEN "R",1,"TRAT:0","FF",40
4320 FOR Z=1 TO
     U-1:U1$=STR$(F1%(Z)):RN$=STR$(F3%(Z)):PJ$=F2$(Z):QQ$=STR$(F6(Z)):PP$
     =STR$ (F7(Z)):PUT 1,Z,10040:PRINT U1$,RN$,QQ$,PP$:NEXT Z:CMD"KILL
     SALE:1":RETURN
4500 REM *** SUBROUTINE FOR FINAL MONTH END RESULTS *************
4505 CLOSE:OPEN "R",1,"P:0","FF",20:GET 1,1,10050:DF=VAL(DF$)
4510 LPRINT CHR$(12):LPRINT:LPRINT TAB(25),"MONTH END RESULTS":LPRINT
     STRING$(80,"*"):
4520 LPRINT "SALES    * ";H6
4530 LPRINT "PURCHASES  * ";DF
4540 LPRINT STRING$(34,"-")
4550 LPRINT "DIFFERENCE *  ";H8-DF:LPRINT CHR$(12)
4560 RETURN
REM *** FILE IGEL EXPRESSIONS ***************************
10000 (5)U1$,(6)CN$,(12)Q$,(15)N$,(6)P2$;
10010 (5)U1$,(15)N$,(6)CN$,(9)TG$,(7)PT$;
10020 (10)D1$,(2)U1$,(1)G$,(5)B1$,(10)Q1$,(15)PJ$,(2)S1$,(6)P1$;
10030 (3)U1$,(15)PJ$,(10)QQ$,(8)P5$;
10040 (2)U1$,(3)RN$,(15)PJ$,(10)QQ$,(8)PP$;
10050 (10)DF$,(10)DE$;
```

APPENDIX C.1

PSEUDO-CODE LISTING

MAIN PROGRAM

```
MAIN MENU (1 - 210)
        Open "INV" : find highest record number
        Diminsion arrays
        Print Main Menu
                make selection
                If selection = 1-9 then GOSUB 1-9
                Else goto top
BACKUP (220 - 355)
        Backup disk number 0
        Backup disk number 1
        END program
ADD NEW INGREDIENTS (500 - 770)
        Commodity # is derived (highest current record number+1)and
        printed
        Enter name
        Enter pounds
        Enter Value ($/lb) save in "INV"
        Print this at top of screen
        Edit, continue, or return to main
                If edit then gosub edit ingredient
                If continue start at add ingredient component
                If main return to MAIN MENU
                else error

INVENTORY FILE EDITOR (780 - 1131)
        Open "INV"
        Enter commodity # to be edited
        If commodity # = "M" then return to main
        Print commodity information to screen
        Print edit menu
        Make appropriate changes
        Save information in INV
        If selection = 6 then return

RATION ADD, DELETE AND EDIT RATION (2000 - 3970)
        Print the different unit choices
        Select unit
        Open units ration file
        If ration addition
                Enter ration id
                Repeat until ration complete
                     enter ingredient
                     enter pound
                     store information
                End repeat
                Calculate % of each ingredient in diet
        Then <S>ave, <R>eview, <M>ain, <C>ontinue, or <E>dit
```

```
        If <S> then save ration
        If <R> then review ration
        If <M> then return to MAIN MENU
        If <C> then goto continuation menu
        If <E> then edit the ingredients and pounds or
               if saved already, edit ingredient and percent

        If editing a ration from the MAIN MENU
               Then enter ration number
               Get ration
               Edit ingredient and percent
               resave
               Return the MAIN MENU


DAILY SALES (4000 - 4280)
        Enter date
        Enter unit #
        Enter type <R>ation or <I>ngredient
        Enter ration or ingredient #
        Enter quantity (pounds) sold
        Enter project # (optional)
        Enter <B>ulk or <S>ack
        If information is correct call subs
                     ration sale or ingredient sale
        Else make correction
        After sub returns save the information in SALE

INGREDIENT SALE (5165 - 5060)
        Open ING
        Open INV
        Get inventory information for ingredient sold
        Subtract the sale from inventory
        Calculate the cost
        Figure markup
        Store back inventory
        Call sub individual ingredients
        Return

RATION SALE SUBROUTINE (5165 - 5651)
        Call ration routine
        Get ration
        Open INV
        Open ING
        Repeat until end of ration
               get ingredient
               get percent in ration
               figure quantity of on ingredient in sale
               subtract quantity from inventory
               figure cost and markup
               call individual ingredients sub

HELP MENU (6000 - 6200)
```

Prints help menu
Calls help subs

UNIT LISTING (6300 - 6610)
Repeat until end of unit file
get ration
print ration #, name
End repeat

PURCHASES (7000 - 7270)
Enter date
Enter ingredient
Enter quantity
Enter value - $/lb
If correct, save
If incorrect, make correction.
(continue or go make to main)

INGREDIENT SUBROUTINE (8400 - 8440)
If unit # < 10 then unit #=0+unit #
Save ingredient information
Return

MONTH-END REPORTS (8800 - 8980)
Print month-end reports to screen
Gosub selection

REPORT MENU (10000 - 10170)
Prints report menu to screen
Check printer status
Gosub selection

TOTAL UNIT RATION LISTING (12100 - 12250)
Gosub ration
Pick unit
Repeat until end of file
print complete ration on line printer
End repeat
Return

PURCHASE PRINTOUT (16000 - 16120)
Print PURC file to line printer
Return

SALES PRINTOUT (17000 - 31345)
Get sale
Print SALE file to line printer
Return

ERROR SUB (40000 - 40100)
MID$ (position) if numberic OK else NOT OK

YES-NO SUB (40300 - 40340)

MID$(position) one if Y or N OK else NOT OK

DATE ERROR SUB (45000 - 45155)
If length =8
If month <=12
If days <=31
If year >82
then OK else NOT OK

FEEDMILL PICTURE (50000 - 50170)
Print graphics feedmill to the screen
Fill IV$ with ingredient names

SALES CORRECTION SUB (51000 - 51480)
Get the sale
Change the incorrect item
Zero out the old sale
Save the new sale
Return

INVENTORY INGREDIENT LISTING (60030 - 60280)
Open INV
Print to screen commodity numbers and names

INDIVIDUAL INGREDIENT PRINT (60300 - 60410)
Open INV
Print to screen an individual ingredient

PURCHASE CORRECTION (61000 - 61310)
Get purchase
Correct purchase
Save the purchase

CORRECTION MENU (62000 - 65000)
Print the Menu
Gosub selection

END PROGRAM

APPENDIX C.2

REPORTS PSEUDO-CODE LISTING

```
MONTHS INDIVIDUAL INGREDIENT USE REPORTS (1-500,AND 1000-1200)
        Fill arrays (ING and TOT)
        Set up index array
        Merge the the arrays into one
        Print by unit the information
MONTHS PROJECT AND RATION NUMBER USE REPORTS (500-1000, AND
        1200-3170)
        Fill arrays
        Set up index array
        Merge ration and project arrays with SALE arrays
        Print the project number results
        Print the ration number results
MONTH-END CLOSEOUT (4000 -  )
        Closeout ING
        Closeout SALE
        Closeout Purch

END PROGRAM
```

APPENDIX D

Profile III+ (hereafter called Profile) database files can be used with the BASIC language. Profile III+ is a data base designed to run on Radio Shack computers. It stores data record by record in variable length random access files, in ASCII format.

Profile III+ is capable of a variety of sorting and searching techniques, and because it is in machine language, these techniques are fast and effieient. However, report production is somewhat limited. For example, if you request a mean from a set of data that includes missing entries, division is by the total number of records, not the number of "valid" records. Generating reports using BASIC can relieve that problem.

The example below is a database set up to handle a sheep carcass show. A description of Profile III+ and its commands are included in the Profile documentation manual (Tandy, 1980C). The database is a one segment file that records information about the carcass and live data. Figure 1 lists the "LAMB" fields, along with their lengths, and the headings

Profile stores the data under the file name, LAMB, 127 ASCII characters per record, in the order the records were entered.

The first step in accessing Profile from BASIC is to set up an index, using Profile. The index is generated by setting up selection and sort criteria. Suppose you asked Profile to generate an index of all sheep in the show with a live weight of over 120 lbs, then sort that list by entry number. Profile would examine the entire LAMB file. Each time a record was encountered where live weight was 120 lbs. or greater, a record would be created in the index file containing the record number (from LAMB), and the entry number. When all valid records are selected and the data compiled in the index, the index is sorted in order of entry number. Now, records from LAMB can be accessed in the order

that the record numbers appear in the index. In this program the, index file is to be sorted by class number. (Tandy, 1980C).

This BASIC program gets its data from two different Profile files. The first is the index file "LAMB/IX1", which is the user index containing the list, sorted by class number, of the records in the file. It is accessed by buffer 1 and has a record length of 5 (The field length plus 3 "house keeping" bytes). See line 45 of the source code.

The second file is "LAMB/KEY", which contains the physical records of the data base. It is accessed through buffer 2 and has a record length of 124, which is the segment length specified when the file was created. You can also determine record lengths of Profile files from the directory.

An important factor to remember is to OPEN files for variable lengths, in this case (2V), (Tandy,1980B). The house keeping refered to above includes the location of desired record and remaining unused characters.

This BASIC program sorts and prints the carcass show results. It allows the information to be entered into the data base and with the capabilities of BASIC, sorts and prints the carcass show results by class and place. Therefore, when the information is entered it does not have to be in any specified order.

USER GUIDE FOR CARCL

First, the data base must be filled. Second when this is completed and all data have been entered, create an index file. This index file is sorted by class number. This index file should be saved on drive 1 along with the Key file. Next mount a TRS-DOS disk containing BASIC (BASIC is absent from most Profile disks). Enter the date and time. Type <BASIC>. When the program asks for number of files, enter **2V**. This is very important. There are no restriction on memory size so just enter. READY, type RUN "CARCL" then enter. Make sure

the printer is set. It requires a 132 column printer. The results and listing follow in Figure 2.

The record numbers in the index are in compressed (hexidecimal) format and must be recovered using the BASIC CVI statement. This converts 2 byte string variables to integers. In the BASIC program the first record accessed is record four. This is because the first three records are for internal housekeeping records.

Field	Heading	Length
1	Last	20
2	First	10
3	City	15
4	Breed	2
5	Class#	3
6	LPlace	1
7	Eartag	5
8	LVWT	3
9	CWT	3
10	Conf	3
11	Maturity	3
12	QScore	3
13	QGrade	3
14	Thickact	3
15	Thick adj	3
16	Legconf	3
17	Intfat	3
18	Y. G.	4
19	Rib eye area	3
20	CPlace	2
21	Rate of Gain	3
22	Dress. Pct.	3

Figure 1: LAMB field assignments

Figure 2: BASIC listing of LAMB

```
5 REM REQUIRES 2V**************************************
10 REM *** PRINTOUTS FOR LAMB*******************
11 POKE 16425,0
12 GOSUB 20:GOTO 42
13 CLS:CLEAR 10000
20 LPRINT"                    EAR    WARM                                RATE
30 LPRINT" EXHIBITORS        BRD  LIVE  TAG  LIVE CAR  CONF  MAT
     QUAL  QUAL  THK   LEG  INT  YG. REA CARC OF DRESS
40 LPRINT"  ADDRESS            PLACE  NO.  WT.  WT.       SCR  GRD
     AT/AJ.   CONF. FAT      PLACE GAIN  %":LPRINT STRING$(132,"*")
41 RETURN
42 CLS:
43 DIM AR$(20,20)
REM ** Open the files ************************************
44 OPEN "R",1,"LAMB0000/IX1",5
45 FIELD1,2ASKY$,2ASQQ$,1ASXX$
46 OPEN "R",2,"LAMB0000/KEY",124
50 FIELD 2,15 AS LT$,10 AS FT$,15 AS AD$,15 AS CT$,2 AS ST$,5 AS ZP$,5
     AS BR$,2 AS CL$,2 AS LP$,5 AS ET$,3 AS LW$,3 AS CW$,3 AS LG$,3 AS
     BF$,3 AS FO$,3 AS LE$,3 AS LM$,3 AS MC$,3 AS MS$,3 AS HS$,4 AS
     YG$,4 AS RE$,2 AS CP$,3 AS RG$,4 AS DP$
70 AA$="%       %%    %%  % % % %   % %  % %  % %  % %  % %  %%  %.##
     /.##   %  %%  %%  %%  %% %%  %##.#"
80 AB$="  %       %
90 AC$="%       % %     % %  % %  %%   % %  %"
95 AD$="  ##.#
96 AE$="###.# ###.#             .##       #.# #.#  #.#    #.# ##.#
100 J=0:I=1:II=0
110 LR=31
120 GET 1,LR
130 RR=CVI(QQ$)
140 A=VAL(KY$):KK$=MID$(KY$,3,1)
150 GET 2,RR
160 IF LR>LOF(1) THEN LPRINT "EOF"
162 IF VAL(CW$)=0 THEN GOTO 165
165 LPRINT "CLASS - ";CL$
180 FOR XX=5 TO LOF(1):GOSUB 3000
181  CLOSE 2:OPEN "R",2,"LAMB0000/KEY",124
182 FIELD 2,15 AS LT$,10 AS FT$,15 AS AD$,15 AS CT$,2 AS ST$,5 AS ZP$,5
     AS BR$,2 AS CL$,2 AS LP$,5 AS ET$,3 AS LW$,3 AS CW$,3 AS LG$,3 AS
     BF$,3 AS FO$,3 AS LE$,3 AS LM$,3 AS MC$,3 AS MS$,3 AS HS$,4 AS
     YG$,4 AS RE$,2 AS CP$,3 AS RG$, 4ASDP$
186 LR=LR+1
190 GET 1,LR
200 RR=CVI(QQ$)
210 GET 2,RR
215 PRINT RR,LT$,LR
220 IF A=VAL(KY$) AND MID$(KY$,3,1)=KK$ THEN GOTO 270 ELSE GOSUB
     3185:GOSUB 500:GOSUB 1000:
```

```
270 A=VAL(KY$):KK$=MID$(KY$,3,1)
280 NEXT XX
286 GOSUB 3000
290 GOSUB 3185
295 GOSUB 500:GOSUB 1000
300 STOP
500 REM PRINT OUT SUBROUTINE
505 DD=1
510 FOR Z=1 TO II
512 IF PEEK(16425)>52 THEN LPRINT CHR$(12):GOSUB 20
515 IF VAL(AR$(Z,7))=0 THEN LPRINT USING
      AC$;AR$(Z,1),AR$(Z,2),AR$(Z,3),AR$(Z,4),AR$(Z,5),AR$(Z,6);:LPRINT "
      CARCASS NOT AVAILABLE":GOTO 530
516 IF VAL(AR$(Z,6))=0 THEN LPRINT USING
      AA$;AR$(Z,1),AR$(Z,2),AR$(Z,3),AR$(Z,4),AR$(Z,5),AR$(Z,6),AR$(Z,7),AR$
      (Z,8), AR$(Z,9),AR$(Z,10),AR$(Z,11),VAL(AR$(Z,12)),VAL(AR$
      (Z,13)),AR$(Z,14),AR$(Z,15),AR$(Z,16),AR$(Z,17),AR$(Z,18),AR$(Z,19):GOT
520 LPRINT USING
      AA$;AR$(Z,1),AR$(Z,2),AR$(Z,3),AR$(Z,4),AR$(Z,5),AR$(Z,6),AR$(Z,7),AR$
      (Z,8),
      AR$(Z,9),AR$(Z,10),AR$(Z,11),AR$(Z,12),AR$(Z,13),AR$(Z,14),AR$(Z,15),A
      R$(Z,1 6),AR$(Z,17),VAL(AR$(Z,7))/VAL(AR$(Z,6))*100
530 LPRINT USING AB$;AR$(Z,18)
540 J=J+VAL(AR$(Z,6))
545 IF VAL(AR$(Z,7))=0 THEN GOTO 660
550 H=H+VAL(AR$(Z,7))
560 K=K+VAL(AR$(Z,8))
570 L=L+VAL(AR$(Z,9))
580 M=M+VAL(AR$(Z,10))
590 N=N+VAL(AR$(Z,11))
600 O=O+VAL(AR$(Z,12))
610 P=P+VAL(AR$(Z,13))
620 Q=Q+VAL(AR$(Z,14))
630 R=R+VAL(AR$(Z,17))
640  IF VAL(AR$(Z,7))<>0 THEN S=S+VAL(AR$(Z,7))/VAL(AR$(Z,6))
650 T=T+VAL(AR$(Z,15))
660 NEXT Z
670 RETURN
1000 REM PRINT AVG
1010 J=J/II:J$=STR$(J)
1020 H=H/I:H$=STR$(H)
1030 K=K/I:K$=STR$(K)
1040 L=L/I:L$=STR$(L)
1050 M=M/I:M$=STR$(M)
1060 N=N/I:N$=STR$(N)
1070 O=O/I:O$=STR$(O)
1080 P=P/I:P$=STR$(P)
1090 Q=Q/I:Q$=STR$(Q)
1100 R=R/I:R$=STR$(R)
1110 S=S/I:S$=STR$(S)
1111 T=T/I
1120 LPRINT STRING$(132,"-")
1130 LPRINT "           BREED AVERAGE(";I;" HEAD)  ";:
```

```
1140 LPRINT USING AE$;J,H,K,L,M,N,O,P,Q,T,R,S*100
1150 LPRINT STRING$(132,"-")
1160 A=VAL(KY$):B=VAL(QQ$)
1165 IF PEEK(16425)=52  OR PEEK(16425)=51 THEN LPRINT CHR$(12)
1170 LPRINT :LPRINT "CLASS - ";CL$
1175 LPRINT
1177 J=0:H=0:K=0:L=0:M=0:N=0:O=0:P=0:Q=0:R=0:T=0:S=0:
1178 FOR BB=1 TO 20:FOR CC=1 TO 18:AR$(BB,CC)="":NEXT CC:NEXT BB
1179 II=0:I=1:
1180 RETURN
3000 REM FILLS THE ARRAYS
3006 IF VAL(AR$(II,7))=0 THEN GOTO 3010
3007 I=I+1:
3010 II=II+1::AR$(II,1)=FT$
3020 AR$(II,2)=LT$
3030 AR$(II,3)=BR$
3040 AR$(II,4)=LP$
3050 AR$(II,5)=ET$
3060 AR$(II,6)=LW$
3070 AR$(II,7)=CW$
3080 AR$(II,8)=LG$
3090 AR$(II,9)=BF$
3100 AR$(II,10)=FO$
3110 AR$(II,11)=LE$
3120 AR$(II,12)=LM$
3130 AR$(II,13)=MC$
3140 AR$(II,14)=MS$
3150 AR$(II,15)=HS$
3160 AR$(II,16)=CP$
3170 AR$(II,17)=RG$
3180 AR$(II,18)=CT$
3184 RETURN
3185 REM SORT BUBBLE
3189 IF II=1 THEN GOTO 3270
3190 FOR Z=1 TO II-1
3200    FOR W=Z+1 TO II
3210     IF VAL(AR$(Z,4)) <= VAL(AR$(W,4)) THEN GOTO 3250
3220      FOR ZZ=1 TO 18
3230       T$=AR$(Z,ZZ):AR$(Z,ZZ)=AR$(W,ZZ):AR$(W,ZZ)=T$
3240      NEXT ZZ
3250   NEXT W
3260 NEXT Z
3270 RETURN
```

A MICROCOMPUTER PROGRAM FOR INVENTORY CONTROL AND BILLING FOR A UNIVERSITY FEED MANUFACTURING FACILITY

by

Coree Lynn Smith

B. S., Kansas State University, 1981

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Animal Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1984

ABSTRACT

This report presents a BASIC language microcomputer program that controls billing and inventory for a feedmill servicing a variety of livestock research units in university Animal Science departments.

A beginning inventory of ingredients, their prices, and total value per item is established. When ingredients are purchased, the total value per item is recomputed and a new price per pound is established.

Rations are identified by unit number and research project. To enter a new ration, the weight of each ingredient is listed, then the percent of each ingredient is atomatically computed. For management purposes, the department receives reports of ingredient and ration use for each unit and project. When a ration sale is entered, the total pounds of the ration is multiplied by the percent of each ingredient. The appropriate amount of each ingredient is subtracted from inventory, and the value of the ration computed.

Reports include all input information as well as month-end reports of production value and inventory. The month-end reports or month-to-date reports are sorted and printed by research unit. Included are total unit sales by ingredient, by project number, and by ration number.

The program runs on a Radio Shack model III or 4 with two disk drives and at least 48K of user memory.